



TUGAS AKHIR – KS 141501

# OJEK *FINDER*: APLIKASI MOBILE BERBASIS KOMUNITAS UNTUK MEMBERIKAN AKSES TRANSPORTASI TAKSI MOTOR KEPADA MASYARAKAT

Aldo Kelvianto Wachyudi  
NRP 5211 100 092

Dosen Pembimbing  
Arif Wibisono, S.Kom., M.Sc.  
Amna Shifia Nisafani, S.Kom., M.Sc.

JURUSAN SISTEM INFORMASI  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2016



FINAL PROJECT - KS 141501

# OJEK FINDER: COMMUNITY BASED MOBILE APPLICATION TO PROVIDE PUBLIC AN ACCESS TO MOTORCYCLE TAXY

ALDO KELVIANTO WACHYUDI  
NRP 5211 100 092

Aldo Kelvianto Wachyudi  
NRP 5211 100 092

Supervisor  
Arif Wibisono, S.Kom., M.Sc.  
Amna Shifia Nisafani, S.Kom., M.Sc.

DEPARTMENT OF INFORMATION SYSTEMS  
Faculty of Information and Technology  
Institut Teknologi Sepuluh Nopember  
Surabaya 2016

## LEMBAR PENGESAHAN

### **OJEK FINDER: APLIKASI MOBILE BERBASIS KOMUNITAS UNTUK MEMBERIKAN AKSES TRANSPORTASI TAKSI MOTOR KEPADA MASYARAKAT**

#### **TUGAS AKHIR**

Disusun untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
pada  
Jurusan Sistem Informasi  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember

Oleh:

**ALDO KELVIANTO WACHYUDI**  
**5211 100 092**

Surabaya, Januari 2016

**KETUA  
JURUSAN SISTEM INFORMASI**

**Dr. Ir. Aris Tjahyanto, M.Kom.**  
**NIP 19650310 199102 1 001**




# **OJEK FINDER: APLIKASI *MOBILE* BERBASIS KOMUNITAS UNTUK MEMBERIKAN AKSES TRANSPORTASI TAKSI MOTOR KEPADA MASYARAKAT**

**Nama Mahasiswa** : Aldo Kelvianto Wachyudi  
**NRP** : 5211 100 092  
**Jurusan** : Sistem Informasi FTIF-ITS  
**Dosen Pembimbing I** : Arif Wibisono, S.Kom., M.Sc.  
**Dosen Pembimbing II** : Amna Shifia Nisafani, S.Kom., M.Sc

## **ABSTRAK**

*Tuntutan akan mobilitas tinggi merupakan salah satu tantangan hidup pada modern ini. Sayangnya kemacetan di jalan raya menghambat mobilitas masyarakat, terutama masyarakat yang menggunakan kendaraan besar seperti mobil. Di sisi lain sepeda motor muncul sebagai kendaraan yang mampu melewati kemacetan pada sela-sela mobil di jalan besar. Sepeda motor juga mampu melewati jalan-jalan sempit yang tidak mampu dilewati mobil. Alhasil banyak masyarakat memilih menggunakan sepeda motor karena fleksibilitasnya cukup tinggi. Seiring dengan perkembangan teknologi, munculah berbagai layanan taksi motor online. Layanan taksi motor online ternyata digemari masyarakat karena mobilitasnya yang cukup tinggi dan memberikan kemudahan untuk memesan taksi motor. Sayangnya kebanyakan supir taksi motor secara online harus terikat pada suatu instansi. Karena keterbatasan instansi tersebut, tidak semua masyarakat dapat menyediakan taksi motor. Padahal potensi dari masyarakat yang bisa merangkap sebagai supir taksi motor cukup banyak. Ojek Finder bertindak sebagai bursa supir taksi motor dan penumpang yang membutuhkan taksi motor tanpa mengharuskan masyarakat sebagai karyawan dari Ojek Finder. Hal ini memungkinkan masyarakat menjadi supir taksi motor hanya ketika memiliki*



*waktu senggang dan tidak menerima penawaran penumpang ketika sedang berhalangan. Ojek Finder juga tidak mematok tarif khusus seperti aplikasi taksi motor online lainnya. Sistem pembayaran diserahkan kepada pasar penawaran dan permintaan, sehingga memungkinkan adanya layanan gratis hingga berbayar. Dengan adanya Ojek Finder masyarakat memiliki akses terhadap layanan transportasi dengan fleksibilitas tinggi yang berbasis komunitas sehingga diharapkan mampu membantu meningkatkan mobilitas masyarakat.*

***Kata Kunci: Taksi roda dua, taksi motor, ride sharing, sharing economy, mobile application, community based, backend as service, customer to customer***




# **OJEK FINDER: COMMUNITY BASED MOBILE APPLICATION TO PROVIDE PUBLIC AN ACCESS TO MOTORCYCLE TAXY**

**Name** : Aldo Kelvianto Wachyudi  
**NRP** : 5211 100 092  
**Department** : Sistem Informasi FTIF-ITS  
**Academic Promotor I** : Arif Wibisono, S.Kom., M.Sc.  
**Academic Promotr II** : Amna Shifia Nisafani, S.Kom., M.Sc

## **ABSTRACT**

*High mobility lifestyle is becoming standard for modern age. However, traffic jam is one of the challenge that hinder citizen mobility, especially car owners. On another side, motorcycle emerges as a good alternative for solving traffic problem. Motorcycle able to operates at narrow streets and sidelines of cars. Those advantages causes a lot of citizen to use motorcycle as a transportation alternative. Some entrepreneur sees this opportunity and creates an e-hailing application for motorcycle taxi. Turns out, citizen loves it. However, not everyone could apply as a motorcycle taxi driver. Some person whose busy for most times can't becomes a driver when they have spare time. We can utilize these potential user so that they can become a motorcycle taxi on their spare time. That's where Ojek Finder comes in. Ojek Finder serves as a place to meet passenger and potential motorcycle taxi driver. This enable users becomes a motorcycle taxi driver only on their spare time, without a strict company rule. Ojek Finder also have bidding system where passenger can choose variety of price that motorcycle taxi driver use. Price range will be from free to whatever users desire. With Ojek Finder, citizen will have access to motorcycle taxi service so they have more flexibility when it comes to choose transportation options. Thus increasing citizen mobility and potentially increasing their income.*

The background of the page is a repeating pattern of lotus flowers inside gears. The lotus flowers are yellow with green outlines, and the gears are light blue. The pattern is arranged in a grid that covers the entire page.

**Keywords:** *Motorcycle Taxy, ride sharing, sharing economy, mobile application, communitiy based, backend as service, customer to customer*



## KATA PENGANTAR

Dengan rasa terima kasih yang mendalam, akhirnya penulis dapat menyelesaikan penulisan skripsi ini tepat pada waktunya. Penulis mengucapkan terimakasih sebesar-besarnya kepada ayahanda dan ibunda yang telah dengan tulus hati membesarkan, mendidik, membiayai dan memberikan dorongan semangat untuk menyelesaikan studi sarjana strata satu.

Pada kesempatan ini juga penulis ingin mengucapkan terima kasih yang sebesar besarnya kepada :

- Bapak Aris Tjahyanto selaku Ketua Jurusan Sistem Informasi ITS selama saya manjalani kuliah di Sistem Informasi ITS. Terima kasih atas semua dukungan fasilitas selama ini.
- Bapak Arif Wibisono selaku pembimbing I dalam proses pengerjaan tugas akhir ini yang selalu bersedia waktunya diganggu untuk bimbingan tugas akhir dan tetap sabar memberikan bimbingan meskipun saya sempat menghilang pada awal-awal pengerjaan.
- Ibu Amna Shifia Nisafani selaku pembimbing II dalam proses pengerjaan tugas akhir ini yang selalu bersedia memberi masukan mengenai analisa dan desain dalam mengerjakan tugas akhir.
- Ibu Nur Aini, Bapak Febriliyan Samopa dan Bapak Andre Parvian selaku penguji I dan penguji II yang bersedia menguji dan memberikan masukan yang berharga terhadap tugas akhir saya.
- Bapak Shaliq selaku dosen wali selama 4 tahun menjalani kuliah di jurusan Sistem Informasi ITS
- Mas Bambang yang telah memudahkan saya dalam menjalani berbagai sidang di laboratorium E-Bisnis.
- Teman-teman jurusan Sistem Informasi ITS, yang menemani saya berjuang di kampus perjuangan. Khususnya saudara Ahmad, Aula, Nisa dan Amril yang memberikan bantuan teknis mengenai pengerjaan laporan tugas akhir.

Penulis menyadari bahwa tugas akhir ini masih banyak kekurangan dan ketidaksempurnaan, untuk itu penulis



mengharapkan saran atas tugas akhir ini yang bersifat membangun guna perbaikan di masa mendatang. Akhir kata, penulis berharap tugas akhir ini dapat bermanfaat bagi kita semua.

Surabaya, 8 Januari 2016

**Penulis**

## DAFTAR ISI

ABSTRAK.....	v
ABSTRACT.....	vii
KATA PENGANTAR.....	ix
DAFTAR ISI.....	xi
DAFTAR GAMBAR.....	xv
DAFTAR TABEL.....	xxi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang Masalah.....	1
1.2 Rumusan Masalah.....	4
1.3 Batasan Masalah.....	4
1.4 Tujuan Tugas Akhir.....	5
1.5 Manfaat Tugas Akhir.....	5
1.6 Relevansi.....	5
BAB II TINJAUAN PUSTAKA.....	7
2.1 Studi sebelumnya.....	7
2.2 Dasar Teori.....	8
2.2.1 Taksi Motor dan Taksi Motor Online.....	9
2.2.2 <i>Mobile Application</i> .....	10
2.2.3 <i>Ride Sharing</i> .....	10
2.2.4 <i>Sharing economy</i> .....	11
2.2.5 Software as a Service.....	11
2.2.6 Crowdsourcing.....	12
BAB III METODOLOGI.....	15
3.1 Tahapan Pelaksanaan Tugas Akhir.....	16
3.1.1 Studi Literatur.....	16
3.1.2 Analisis Kebutuhan Perangkat Lunak.....	16
3.1.3 Desain Aplikasi.....	17
3.1.4 Pengkodean Aplikasi.....	17
3.1.5 Pengujian Aplikasi.....	17
3.1.6 Penyusunan Laporan Tugas Akhir.....	18
BAB IV ANALISIS DAN DESAIN SISTEM “OJEK FINDER”.....	19
4.1 Benchmarking.....	19
4.1.1 Gojek.....	19
4.1.2 GrabTaxi.....	22

4.1.3 Uber .....	26
4.1.4 <i>Requirement Specification</i> .....	30
4.2 Analisis dan Desain “Ojek Finder” .....	31
4.2.1 Deskripsi Aplikasi “Ojek Finder” .....	31
4.2.2 Proses Bisnis Ojek Finder.....	32
4.2.3 Kebutuhan Fungsional Sistem .....	34
4.2.4 Kebutuhan non-fungsional sistem .....	35
4.2.5 Aktor-Aktor Dalam Sistem.....	35
4.2.6 Use Case .....	36
4.2.7 Use Case Description.....	43
4.2.8 Sequence Diagram.....	70
4.2.9 Class Diagram .....	70
4.2.10 Entity Relationship Diagram .....	75
BAB V IMPLEMENTASI.....	77
5.1 Lingkungan Implementasi .....	77
5.2 Struktur Direktori.....	78
5.3 Implementasi Ojek Finder.....	78
5.3.1 Implementasi Bergabung Dengan Komunitas.....	79
5.3.2 Implementasi melihat lokasi keberadaan semua pengguna.....	82
5.3.3 Implementasi Membuat Permintaan Tumpangan .....	84
5.3.4 Implementasi Mengajukan Penawaran Taksi Motor.....	88
5.3.5 Implementasi Menyetujui Tawarkan Taksi Motor 90	
5.3.6 Implementasi Mengakhiri Persetujuan Taksi Motor.....	93
5.3.7 Implementasi Memberikan Rating dan Review .....	95
5.3.8 Implementasi Mengaktifasi dan Deaktivasi Anggota.....	96
5.3.9 Implementasi Pembuatan Admin Baru.....	98
5.3.10 Implementasi Pembuatan Komunitas dan Super Admin Komunitas .....	99
5.4 Pengujian.....	99



5.4.1	Uji Coba Fungsional.....	99
5.4.2	User Acceptance Test.....	100
<b>BAB VI HASIL DAN PEMBAHASAN .....</b>		<b>103</b>
6.1	Hasil Pengujian .....	103
6.1.1	Uji Coba Fungsional.....	103
6.1.2	Hasil Pengujian User Acceptance Test .....	104
6.2	Analisa Pengujian .....	106
6.2.1	Analisa Uji Coba Fungsional .....	106
6.2.2	Analisis Pengujian User Acceptance Test...	107
<b>BAB VII KESIMPULAN DAN SARAN.....</b>		<b>109</b>
7.1	Kesimpulan .....	109
7.2	Saran .....	110
<b>DAFTAR PUSTAKA .....</b>		<b>111</b>
<b>BIODATA PENULIS .....</b>		<b>114</b>

## DAFTAR GAMBAR

Gambar 1 Road Map Lab E-Bisnis .....	6
Gambar 2 Model <i>Iterative Waterfall</i> .....	15
Gambar 3 Tahap Pengerjaan Tugas Akhir.....	16
Gambar 4 Menu Utama Gojek.....	20
Gambar 5 Menu Pengemudi Di Sekitar Pengguna .....	21
Gambar 6 Tampilan Bila Terjadi Persetujuan Antara Penumpang dan Supir Gojek .....	22
Gambar 7 Pilihan Kendaraan Dalam GrabTaxi.....	23
Gambar 8 Tampilan Pengemudi Di Sekitar Pengguna .....	24
Gambar 9 Tampilan Pemilihan Lokasi Keberangkatan dan Tujuan .....	25
Gambar 10 Tampilan Penjemputan Penumpang Oleh Supir .	26
Gambar 11 Penambahan Informasi Detil Transaksi .....	27
Gambar 12 Konfirmasi Pemesanan Taksi.....	28
Gambar 13 Fitur Pembagian Ongkos Kirim .....	29
Gambar 14 Pemberian Rating dan Review .....	30
Gambar 15 Pengajuan Komunitas Baru.....	32
Gambar 16 Pengajuan Verifikasi Pengguna .....	32
Gambar 17 Verifikasi Pengguna Oleh Admin.....	33
Gambar 18 Permintaan Tumpangan Taksi Motor .....	33
Gambar 19 Use Case Super Admin Komunitas.....	38
Gambar 20 Use Case Admin Komunitas .....	39
Gambar 21 Use Case Pengguna.....	40
Gambar 22 Use Case Penumpang .....	41
Gambar 23 Use Case Supir Taksi Motor .....	42
Gambar 24 Use Case Root Admin.....	43
Gambar 25 Class Diagram Ojek Finder.....	72
Gambar 26 Class Diagram Admin Komunitas .....	73
Gambar 27 Class Diagram Super Admin Komunitas.....	74
Gambar 28 Skema Ojek Finder.....	75
Gambar 29 Sturktur Direktori Ojek Finder.....	78
Gambar 30 Implementasi Bergabung Dengan Komunitas ....	79
Gambar 31 SS Bergabung Dengan Komunitas.....	80
Gambar 32 Implementasi Bergabung Dengan Komunitas 2 .	81
Gambar 33 SS Bergabung Dengan Komunitas 2.....	81



Gambar 34 Implementasi Melihat Lokasi Keberadaan Semua Pengguna.....	82
Gambar 35 Implementasi Melihat Lokasi Keberadaan Semua Pengguna 2.....	83
Gambar 36 SS Keberadaan Semua Pengguna .....	83
Gambar 37 Implementasi Membuat Permintaan Tumpangan 84	
Gambar 38 SS Implementasi Membuat Permintaan Tumpangan .....	85
Gambar 39 Implementasi Membuat Permintaan Tumpangan 2 .....	86
Gambar 40 SS Membuat Permintaan Tumpangan.....	87
Gambar 41 SS Membuat Permintaan Tumpangan 3.....	88
Gambar 42 Implementasi Pengajuan Penawaran Taksi Motor .....	89
Gambar 43 SS Penawaran Taksi Motor.....	89
Gambar 44 Implementasi Pengajuan Penawaran Taksi Motor .....	90
Gambar 45 Implementasi Persetujuan Taksi Motor .....	90
Gambar 46 SS Menyetujui Penawaran Taksi Motor .....	91
Gambar 47 Implementasi Persetujuan Taksi Motor 2 .....	91
Gambar 48 SS Menyetujui Tawaran Taksi Motor 2 .....	92
Gambar 49 SS Menyetujui Tawaran Taksi Motor 3 .....	93
Gambar 50 Implementasi Mengakhiri Persetujuan Taksi Motor .....	94
Gambar 51 SS Implementasi Pembatalan Taksi Motor.....	94
Gambar 52 Implementasi Mengakhiri Persetujuan Taksi Motor 2.....	95
Gambar 53 Implementasi Memberikan Rating dan Review..	95
Gambar 54 SS Memberikan Rating dan Review .....	96
Gambar 55 Implementasi Memberikan Rating dan Review	296
Gambar 56 Implementasi Mengaktifasi dan Deaktifasi Anggota.....	97
Gambar 57 SS Aktivasi dan Deaktivasi Anggota.....	97
Gambar 58 Implementasi Pembuatan Admin Baru .....	98
Gambar 59 SS Pembuatan Admin Baru.....	98
Gambar 60 Implementasi Pembuatan Komunitas dan Super Admin Komunitas .....	99



Gambar 61 Implementasi Pembuatan Komunitas dan Super Admin Komunitas 2 .....	99
Gambar 62 SD Pengguna Melakukan Login .....	1
Gambar 63 SD Pengguna Melakukan Logout .....	2
Gambar 64 SD Pengguna Melihat Lokasi Pengguna Lain .....	3
Gambar 65 SD Pengguna Melihat Lokasi Pengguna Di Sekitarnya .....	4
Gambar 66 SD Pengguna Membatalkan Transaksi Confirm... ..	5
Gambar 67 SD Pengguna Memberikan Review Rating .....	7
Gambar 68 SD Pengguna Mendaftar Baru .....	7
Gambar 69 SD Pengguna Mengedit Profil .....	8
Gambar 70 SD Admin Komunitas Login .....	9
Gambar 71 SD Admin Komunitas Logout .....	10
Gambar 72 SD Admin Komunitas Mengubah Status Verifikasi Member .....	11
Gambar 73 SD Penumpang Membuat Pemesanan Taksi Motor Baru .....	12
Gambar 74 SD Penumpang Menyetujui Penawaran Taksi Motor .....	13
Gambar 75 SD Super Admin Komunitas Create New Admin .....	14
Gambar 76 SD Super Admin Komunitas Log Out .....	15
Gambar 77 SD Super Admin Komunitas Login .....	16
Gambar 78 SD Supir Taksi Motor Memberikan Tawaran Tumpangan Taksi Motor .....	17
Gambar 79 CD CommunityAdminApplication .....	1
Gambar 80 CD MemberItemViewAdapter .....	1
Gambar 81 CD CommunityAdmin .....	2
Gambar 82 CD Member .....	2
Gambar 83 CD ModelUserCommunity .....	3
Gambar 84 CD UserC .....	3
Gambar 85 CD UserCommunity .....	3
Gambar 86 CD LoginActivity .....	4
Gambar 87 CD MainActivity .....	4
Gambar 88 CD MemberDetailFragment .....	4
Gambar 89 CD ViewHolder .....	5
Gambar 90 CD Constants .....	5
Gambar 91 CD OjekFinderApplication .....	5

Gambar 92 CD ConfirmAdapter.....	6
Gambar 93 CD LandingViewPagerAdapter .....	6
Gambar 94 CD MainViewPagerAdapter .....	7
Gambar 95 CD PassengerAdapter .....	7
Gambar 96 CD ReviewRatingAdapter .....	8
Gambar 97 CD RiderAdapter .....	8
Gambar 98 CD Community .....	9
Gambar 99 CD Confirm.....	9
Gambar 100 CD ConfirmFlag .....	10
Gambar 101 CD Flag.....	10
Gambar 102 CD ModelConfirm .....	11
Gambar 103 CD ModelPassengerRequest.....	11
Gambar 104 CD ModelReviewRating.....	12
Gambar 105 CD ModelRiderRequest.....	12
Gambar 106 CD PassengerRequest .....	13
Gambar 107 CD ReviewRating .....	13
Gambar 108 CD RiderRequest .....	14
Gambar 109 CD User.....	14
Gambar 110 CD UserCommunity .....	15
Gambar 111 CD ConfirmedListFragment .....	16
Gambar 112 CD EditProfileActivity .....	17
Gambar 113 CD LandingActivity.....	17
Gambar 114 CD LandingFragment .....	17
Gambar 115 CD LocationDetailActivity .....	18
Gambar 116 CD LoginFragment .....	18
Gambar 117 CD MainActivity.....	19
Gambar 118 CD MainFragment .....	20
Gambar 119 CD MapsActivity .....	20
Gambar 120 CD NewRequestActivity.....	21
Gambar 121 CD PassengerListFragment.....	22
Gambar 122 CD ProfileActivity .....	23
Gambar 123 CD RequestCommunityActivity .....	24
Gambar 124 CD ReviewRatingActivity .....	24
Gambar 125 CD SignUpFragment.....	25
Gambar 126 CD WalkthroughOneFragment.....	25
Gambar 127 CD WalkthroughTwoFragment .....	25
Gambar 128 CD BlankFragment .....	25



Gambar 129 CD CircleTransform .....	26
Gambar 130 CD Constants .....	26
Gambar 131 CD DrawableUtil .....	26
Gambar 132 CD Earth .....	26
Gambar 133 CD FontUtil.....	27
Gambar 134 CD GeneratePicture .....	27
Gambar 135 CD OFDialog .....	27
Gambar 136 CD PictureUtil.....	27
Gambar 137 CD RandomStringUUID .....	28
Gambar 138 CD RenderScriptBlur .....	28
Gambar 139 CD SoftKeyboardUtil .....	28
Gambar 140 CD StackBlur .....	28
Gambar 141 CD Toaster .....	29
Gambar 142 CD TypefaceCache .....	29
Gambar 143 CD Admin .....	29
Gambar 144 CD AdminDetailActivity .....	30
Gambar 145 CD AdminDetailFragment.....	30
Gambar 146 CD LoginActivity .....	30
Gambar 147 CD NewAdminActivity .....	31
Gambar 148 CD SuperAdminApplication .....	31
Gambar 149 CD EncryptionUtil .....	31



## DAFTAR TABEL

Tabel 1 Perbandingan Ojek Finder Dengan Aplikasi Sejenis..	8
Tabel 2 Perbandingan Fitur Gojek, GrabTaxi, dan Uber .....	30
Tabel 3 Tabel Kebutuhan Fungsional .....	34
Tabel 4 Tabel Kebutuhan Non-Fungsional .....	35
Tabel 5 Tabel Aktor .....	35
Tabel 6 Tabel Pemetaan FR dan UC .....	36
Tabel 7 UCD Pendaftaran Pengguna .....	43
Tabel 8 UCD Log In Aplikasi Ojek Finder .....	44
Tabel 9 UCD Mengelola Profil .....	46
Tabel 10 UCD Deaktifasi Akun .....	47
Tabel 11 UCD Melihat Profil Pengguna Lain .....	48
Tabel 12 UCD Melihat Lokasi Seorang Pengguna .....	48
Tabel 13 UCD Melihat Lokasi Seluruh Pengguna Terdekat .	49
Tabel 14 UCD mengajukan Permintaan Komunitas Baru .....	50
Tabel 15 UCD Membatalkan Transaksi .....	51
Tabel 16 UCD Memberikan Rating dan Review .....	52
Tabel 17 UCD Log Out Ojek Finder .....	53
Tabel 18 UCD Memberikan Tawaran Tumpangan Taksi Motor .....	53
Tabel 19 UCD Membuat Pemesanan taksi Motor Baru .....	54
Tabel 20 UCD Menyetujui Penawaran Taksi Motor .....	56
Tabel 21 UCD Log In Super Admin Komunitas .....	57
Tabel 22 UCD Mengubah Status Akun Admin .....	58
Tabel 23 UCD Mengelola Komunitas .....	59
Tabel 24 UCD Menambah Admin Baru .....	59
Tabel 25 UCD Log Out Aplikasi Super Admin Komunitas ..	60
Tabel 26 UCD Log In Ojek Finder Admin .....	61
Tabel 27 UCD Mengubah Status Verifikasi Member Komunitas .....	62
Tabel 28 UCD Deaktifasi Member Komunitas .....	63
Tabel 29 UCD Mengelola Akun Admin .....	64
Tabel 30 UCD Mengelola Komunitas .....	65
Tabel 31 UCD Log Out Aplikasi “Ojek Finder Admin” .....	66
Tabel 32 UCD Log In Root Admin .....	66
Tabel 33 UCD Mengelola Status Komunitas .....	67
Tabel 34 UCD Mengelola Super Admin Komunitas .....	68

Tabel 35 UCD Menerima Pengajuan Komunitas Baru.....	69
Tabel 36 UCD Log Out Root Admin.....	70
Tabel 37 Perangkat Keras Ojek Finder .....	77
Tabel 38 Perangkat Lunak Ojek Finder .....	77
Tabel 39 Format Test Case .....	100
Tabel 40 Daftar Test Case.....	100
Tabel 41 Format UAT.....	102
Tabel 42 Hasil Uji Coba Fungsional.....	103
Tabel 43 Hasil UAT.....	104
Tabel 44 Komentar UAT .....	105
Tabel 45 TC01 Sign Up .....	1
Tabel 46 TC02 Login.....	2
Tabel 47 TC03 Logout.....	3
Tabel 48 TC04 Mengganti Nama dan No. HP.....	3
Tabel 49 TC05 Mengganti Profile Picture .....	5
Tabel 50 TC06 Bergabung ke Komunitas .....	6
Tabel 51 TC07 Keluar Komunitas.....	7
Tabel 52 TC08 Melihat Keberadaan Semua Pengguna .....	9
Tabel 53 TC09 Memberikan Review dan Rating .....	10
Tabel 54 TC10 Membatalkan Transaksi.....	11
Tabel 55 TC11 Membuat Permintaan Tumpangan Baru.....	13
Tabel 56 TC12 Menyetujui Penawaran Taksi Motor .....	14
Tabel 57 Membuat Penawaran Taksi Motor.....	14
Tabel 58 Mengaktivasi dan Deaktivasi Pengguna .....	16
Tabel 59 TC15 Membuat Admin Baru .....	16
Tabel 60 TC16 Mengaktivasi dan Deaktivasi Admin Komunitas.....	18



# BAB I PENDAHULUAN

Pada bab ini, akan dijelaskan tentang Latar Belakang Masalah, Perumusan Masalah, Batasan Masalah, Tujuan Tugas Akhir, dan Relevansi serta Manfaat Kegiatan Tugas Akhir.

## 1.1 Latar Belakang Masalah

Kemacetan bukanlah hal yang baru di Indonesia. Salah satu penyebabnya adalah pertumbuhan jalan besar tidak sebanding dengan peningkatan jumlah mobil. Menurut data dari Dinas Pekerjaan Umum dan Perumahan Rakyat (Dinas PU) jumlah jalan yang ada hanya 60 persen dari total kebutuhan yang sebenarnya. Di Jakarta saja pertumbuhan mobil mencapai 12 persen pertahunnya. Sementara pertumbuhan jalan hanya 0,01 persen [1]. Sehingga pada tahun 2014 Dinas PU mengeluarkan data bahwa kemacetan terjadi di berbagai tempat terutama pada kota-kota besar [2].

Pengguna sepeda motor memiliki keuntungan dalam hal ini. Karena ukurannya yang lebih kecil dari mobil, sepeda motor mampu melewati jalan-jalan sempit serta sela mampu melewati sela-sela mobil dalam kemacetan di jalan besar. Fleksibilitas yang lebih ini cocok sebagai transportasi untuk dua orang dan transportasi barang yang berukuran kecil namun kurang cocok sebagai transportasi barang dengan ukuran dan jumlah besar. Kelebihan sepeda motor ini dilirik oleh pengusaha dengan menggabungkan teknologi *mobile* dan sepeda motor untuk menyediakan layanan taksi motor *online*.

Keberadaan taksi motor sendiri sudah ada sejak tahun 1970 namun pemesanan yang dilakukan melalui ponsel pintar baru dipopulerkan sejak tahun 2011 [3]. Ternyata masyarakat menyambut baik konsep taksi motor *online*, hingga diliput oleh berbagai TV nasional [4]. Bahkan jumlah pengunduh aplikasi layanan taksi motor *online* pada November 2015



sudah melebihi 240 ribu pengguna [5]. Layanan taksi motor juga diyakini berbagai tokoh seperti Gubernur DKI Jakarta Basuki Tjahaja, Menteri Komunikasi dan Informatika Rudiantara, hingga Presiden Republik Indonesia Joko Widodo, mampu mengurai kemacetan di jalan raya [6] [7].

Sayangnya setiap instansi dari supir taksi motor *online* yang sudah ada seperti Gojek dan GrabTaxi memiliki kapasitas dan regulasi yang membatasi ketersediaan taksi motor [8] [9]. Supir taksi motor di layanan taksi motor *online* pun memiliki tuntutan dan jadwal kerja yang tetap, masyarakat yang ingin menyediakan taksi motor namun tidak memiliki waktu untuk *full-time* akhirnya tersisihkan [4]. Sehingga tidak semua masyarakat dapat menjadi supir taksi motor. Padahal masih banyak masyarakat yang berpotensi menjadi supir taksi motor. Karena hal ini penulis mengusulkan sebuah aplikasi *mobile* bernama Ojek Finder.

Aplikasi Ojek Finder bertindak sebagai perantara antara masyarakat yang ingin menjadi supir taksi motor, yang nantinya tergabung dalam komunitas, dan calon penumpang yang membutuhkan taksi motor. Perbedaan yang mendasar dari Ojek Finder dengan penyedia layanan taksi motor *online* yang lain adalah supir taksi motor tidak terikat dengan Ojek Finder, hal ini memungkinkan supir taksi motor melayani permintaan penumpang hanya di waktu senggang dan tidak menerima permintaan ketika sedang ada halangan. Ojek Finder juga tidak memasang tarif khusus untuk layanan taksi motor, supir taksi motorlah yang memasang tarif sendiri dan calon penumpang yang memilih tarif yang paling sesuai dengan calon penumpang.

Kelemahan Ojek Finder dibanding Gojek, GrabTaxi, dan instansi taksi motor *online* lainnya adalah keamanan penumpang Ojek Finder. Karena supir taksi motor Ojek Finder tidak terikat dengan Ojek Finder maka tidak ada standarisasi layanan taksi motor, sehingga pelayanan dari setiap taksi

motor berbeda-beda. Untuk menanggulangi hal ini Ojek Finder menyarankan pengguna untuk bergabung dengan komunitas. Komunitas yang dimaksud adalah sebuah kelompok sosial dari beberapa individu yang memiliki ketertarikan yang sama, baik memiliki ikatan satu instansi, memiliki domisili yang sama, atau memiliki kepercayaan yang sama [10]. Contoh komunitas adalah komunitas jurusan Sistem Informasi, komunitas masyarakat Gebang, atau komunitas pecinta kucing.

Masyarakat yang ingin tergabung dengan Ojek Finder bisa langsung mendaftarkan diri menjadi pengguna melalui aplikasi Ojek Finder, namun untuk memverifikasi bahwa pengguna adalah masyarakat yang sudah terdaftar dalam suatu komunitas yang saling mengenal satu sama lain, pengguna bisa mendaftar kepada Admin dari suatu komunitas [11]. Hal ini semata-mata untuk meningkatkan kepercayaan antara taksi motor dan penumpang dan menjadi pertimbangan pengguna dalam memilih pengguna lain. Ojek Finder akan membantu mempertemukan taksi motor dan penumpang dalam radius lima kilometer. Calon penumpang yang membutuhkan tumpangan akan memberitahu lokasi dan detail terkait penjemputan lalu Ojek Finder akan menyiarkan permintaan tumpangan itu kepada supir taksi motor yang tersedia. Selanjutnya para supir taksi motor akan mengirimkan tarifnya masing-masing. Bila calon penumpang setuju dengan salah satu penawaran supir taksi motor maka akan terjadi akad transportasi dimana supir taksi motor akan menjemput penumpang dan mengantarkan penumpang ke tempat tujuan. Baik penumpang maupun supir taksi motor bisa memberikan ulasan dan *rating* sebagai masukan dari pelayanan kedua belah pihak.

Ojek Finder berpotensi untuk meningkatkan penghasilan masyarakat yang ingin menyediakan layanan taksi motor kepada calon penumpang. Ojek Finder akan menampung data masyarakat yang membutuhkan transportasi taksi motor dan



memberitahu kepada masyarakat yang memungkinkan menyediakan layanan taksi motor. Sehingga Ojek Finder bisa memberikan layanan transportasi yang memiliki fleksibilitas tinggi dalam membantu mobilitas masyarakat.

## 1.2 Rumusan Masalah

Permasalahan yang diangkat dalam tugas akhir ini adalah bagaimana bentuk aplikasi *mobile* berbasis komunitas yang dapat membantu masyarakat mengakses moda transportasi yang dapat mengakomodasi berbagai tantangan transportasi pada berbagai jenis jalan di perkotaan?

## 1.3 Batasan Masalah

Batasan permasalahan dalam tugas akhir ini adalah :

1. Pengguna pada layanan ini memberikan data yang valid mengenai dirinya ke dalam sistem saat melakukan pendaftaran.
2. Aplikasi berasumsi bahwa semua supir taksi motor memiliki kelengkapan dan izin surat kendaraan yang lengkap.
3. Aplikasi berasumsi bahwa semua supir taksi motor dan penumpang tidak memiliki intensi untuk melakukan tindakan kriminal.
4. Aplikasi yang dikembangkan tidak mencakup metode pembayaran secara *online* atau elektronik.
5. Baik penumpang maupun supir taksi motor berada pada satu komunitas dimana semua anggotanya sudah tervalidasi dan terverifikasi.
6. Pengguna layanan baik supir taksi motor maupun penumpang memberikan *review* dan *rating* yang sesuai dengan keadaan sebenarnya tanpa niat untuk memanipulasi data.



7. Aplikasi membutuhkan koneksi internet dan GPS untuk menggunakan layanan sistem.
8. Aplikasi hanya dikembangkan hanya untuk aplikasi Android dengan versi terendah 4.0.

#### **1.4 Tujuan Tugas Akhir**

Tujuan dari pengerjaan tugas akhir ini adalah membuat aplikasi *mobile* berbasis komunitas sehingga komunitas untuk meningkatkan fleksibilitas transportasi.

#### **1.5 Manfaat Tugas Akhir**

Manfaat dari pengerjaan tugas akhir ini adalah:

1. Membantu masyarakat mengakses moda transportasi yang dapat mengakomodasi berbagai tantangan transportasi pada berbagai jenis jalan di perkotaan
2. Memberikan potensi penghasilan tambahan bagi komunitas sebagai supir taksi motor

#### **1.6 Relevansi**

Relevansi penelitian tugas akhir ini terhadap penelitian area sistem informasi berada pada area perencanaan proses e-business dengan topik perencanaan e-business sebagaimana digambarkan pada Gambar 1. Selain itu beberapa mata kuliah terkait pada penelitian ini adalah Algoritma Pemrograman 2, Manajemen Basis Data, Rekayasa Kebutuhan Perangkat Lunak, Rancang Bangun Perangkat Lunak, Pemrograman Perangkat Bergerak, dan Tata Tulis Ilmiah.

## Pohon Penelitian Laboratorium E-Bisnis



**Gambar 1 Road Map Lab E-Bisnis**

## BAB II TINJAUAN PUSTAKA

Pada bagian ini akan dibahas mengenai hasil studi penelitian sebelumnya dan teori-teori yang mendukung dalam pengerjaan tugas akhir.

### 2.1 Studi sebelumnya

Aplikasi Ojek Finder merupakan aplikasi pencarian taksi motor *online* berbasis *crowdsourcing* yang menganut dalam sistem *Sharing Economy* [12]. Dikatakan *crowdsourcing* karena calon penumpang akan menggunakan masyarakat disekitarnya dalam radius lima kilometer untuk mencari supir taksi motor. Disebut *Sharing Economy* karena supir taksi motor rata-rata menggunakan sepeda motor sendiri, yang termasuk dalam aset pribadi.

Ada beberapa aplikasi yang mirip seperti Ojek Finder seperti GrabTaxi, Blu Jek, dan GO-JEK. Semuanya memungkinkan calon penumpang untuk memesan taksi motor menggunakan perangkat ponsel cerdas. Baik GrabTaxi, Blu Jek, dan GO-JEK telah memiliki jumlah pengguna lebih dari 240 ribu orang. Hal ini menunjukkan bahwa aplikasi taksi motor *online* dapat diterima oleh sebagian besar masyarakat Indonesia. Hingga akhir tahun 2015 aplikasi taksi motor *online* telah membuat ribuan orang mendapatkan pekerjaan sebagai supir taksi motor. Supir taksi motor tersebut merupakan masyarakat yang sudah terikat menjadi karyawan tiap instansi tersebut.

Perbedaan yang utama dengan dengan aplikasi taksi motor seperti GrabTaxi dan GOJEK adalah Ojek Finder tidak mengharuskan supir taksi motor untuk terikat dengan Ojek Finder. Hal ini bertujuan untuk mengoptimalkan masyarakat yang ingin melayani kebutuhan transportasi taksi motor pada waktu tertentu saja. Perbedaan lainnya adalah Ojek Finder tidak memasang tarif khusus untuk setiap layanan taksi motor,



namun menyerahkan tarif kepada supir taksi motor dan persetujuan calon penumpang. Sistem pembayaran ini memungkinkan tarif motor yang disediakan gratis hingga berbayar, tergantung persetujuan dari dua belah pihak.

**Tabel 1 Perbandingan Ojek Finder Dengan Aplikasi Sejenis**

	Gojek	GrabTaxi	Ojek Finder
Pembagian Hasil untuk Kontrak Kerja	20% untuk perusahaan 80% untuk supir	10% untuk perusahaan 90% untuk supir	100% untuk komunitas, bagi hasil dengan supir bergantung kesepakatan dengan komunitas supir berafiliasi
Pendaftaran	Kapasitas dan jadwal rekrutmen tergantung perusahaan	Kapasitas dan jadwal rekrutmen tergantung perusahaan	Kapasitas dan jadwal rekrutmen tergantung Komunitas
Tarif	Rp. 2000 per kilometer. Minimal Rp. 15.000	Rp. 3000 per kilometer. Minimal Rp. 25.000	Tarif per kilometer ditentukan oleh anggota komunitas
Supir Taksi Motor	Supir taksi motor Gojek	Supir taksi motor GrabTaxi	Supir yang terdaftar di dalam komunitas saja
Verifikasi	Perusahaan Gojek	Perusahaan GrabTaxi	Admin Komunitas

## 2.2 Dasar Teori

Pada bagian ini akan dijelaskan dasar-dasar teori yang digunakan pada Aplikasi Ojek Finder, yaitu Taksi Motor dan Taksi Motor *Online, Mobile Application, Ride Sharing, Sharing Economy, Software as a Service, Crowdsourcing*.

### 2.2.1 Taksi Motor dan Taksi Motor Online

Taksi motor, ojek atau ojek adalah transportasi umum informal di Indonesia yang berupa sepeda motor atau sepeda, namun lebih lazim berupa sepeda motor [3]. Disebut informal karena keberadaannya tidak diakui pemerintah dan tidak ada izin untuk pengoperasiannya. Penumpang taksi motor seharusnya hanya satu orang tapi tidak jarang juga penumpang berjumlah dua orang atau lebih. Tarif taksi motor ditentukan dengan tawar menawar antar calon penumpang dan supir taksi motor. Setelah terjadi kesepakatan harga antar calon penumpang dan supir taksi motor maka supir taksi motor akan mengantarkan ke tempat yang dituju penumpang.

Taksi motor banyak digunakan oleh penduduk kota-kota besar seperti Jakarta atau Surabaya karena kelebihanannya yang dapat menggunakan sela-sela jalan dalam kemacetan di jalan raya. Taksi motor juga dapat menjangkau daerah-daerah dengan gang-gang yang sempit dan sulit dilalui oleh mobil. Karena peminatnya yang banyak ditambah dengan menjamurnya ponsel pintar, munculah berbagai layanan taksi motor *online* [4].

Taksi motor *online* atau *ojek online* adalah layanan pemesanan taksi motor melalui ponsel cerdas. Dengan menggunakan aplikasi yang harus diunduh terlebih dahulu, pengguna menggunakan aplikasi sebagai antar muka dengan supir taksi motor. Setiap pemesanan taksi motor aplikasi akan menyiarkan permintaan pesanan taksi motor kepada setiap supir taksi motor terdekat.

Kelemahan dari pemesanan taksi motor *online* konvensional seperti Gojek dan GrabTaxi adalah supir taksi motor terikat kontrak kerja dengan instansi tersebut. Gojek dan GrabTaxi mewajibkan supir taksi motornya *full-time* sebagai supir taksi motor. Sehingga masyarakat yang hanya sanggup menjadi supir taksi motor di waktu senggang tidak dapat mendaftar ke Gojek, GrabTaxi, dan taksi motor *online* lainnya.



Aplikasi Ojek Finder memungkinkan pengguna menjadi supir taksi motor untuk melayani permintaan penumpang hanya ketika waktu senggang dan tidak menerima permintaan ketika sedang ada halangan. Ojek Finder juga tidak memasang tarif khusus untuk layanan taksi motor, supir taksi motorlah yang memasang tarif sendiri dan calon penumpang yang memilih tarif yang paling sesuai dengan calon penumpang.

### **2.2.2 Mobile Application**

*Mobile application* (Aplikasi pada perangkat berjalan) merupakan sebuah perangkat lunak yang didesain khusus untuk dioperasikan pada perangkat berjalan seperti *smartphone* dan *tablet*. *Mobile application* dapat memutar musik, menampilkan gambar, dan melakukan operasi komputasional lainnya dalam perangkat berjalan [13]. Kelebihan ini membuat popularitas *mobile application* meningkat hingga mencapai jumlah pengunduh 102 miliar buah pada tahun 2013 [14].

Ojek Finder menggunakan *mobile application* sebagai basis perangkat lunak karena kelebihanannya yang dapat melakukan akses internet untuk mengakses data pengguna, memberikan notifikasi secara *real-time*, serta mengindikasikan lokasi pengguna yang sangat diperlukan untuk transaksi dalam Ojek Finder.

### **2.2.3 Ride Sharing**

*Ridesharing* atau sering juga disebut Manajemen Permintaan Transportasi, Transportasi Alternatif, dan Mobilitas merupakan sebuah metode berbagi kendaraan pribadi untuk mengurangi emisi dari transportasi [15]. Konsep *ridesharing* sudah diterapkan pada aturan mobil pribadi *three-in-one* di Jakarta, dan transportasi publik seperti angkutan kota.



Ojek Finder mengakomodasi *ridesharing* untuk sepeda motor dimana penumpang akan memanfaatkan supir taksi motor yang tersedia untuk mengantarkannya sampai tujuan.

### **2.2.4 Sharing economy**

Sharing economy adalah fenomena berbagi aset layanan maupun barang berbasis *peer-to-peer* yang biasanya terkoordinasi melalui komunitas *online*. Wujud *Sharing economy* bermacam-macam seperti menggunakan teknologi informasi untuk menyediakan informasi untuk individual, korporasi, organisasi non-profit, dan pemerintahan yang bertujuan untuk distribusi ulang, dan berbagi berbagai barang dan layanan. Hal tersebut memungkinkan sebuah individu atau organisasi baik yang mengejar profit ataupun non-profit untuk berbagi kelebihan yang dimilikinya dalam barang, pengetahuan ataupun jasa [16]. Contoh *Sharing economy* adalah AirBnb yaitu layanan berbagi tempat tinggal pribadi sebagai penginapan [17]. Contoh lain adalah layanan berbagi kendaraan yang bernama Gojek [8]. Layanan tersebut membantu mengurangi jumlah kendaraan di jalanan serta mengurangi polusi dengan cara pemakaian mobil bersama.

Aplikasi Ojek Finder memanfaatkan sepeda motor milik pribadi supir taksi motor agar digunakan sebagai transportasi dari pengguna yang membutuhkan tumpangan. Ojek Finder sendiri tidak menyediakan sepeda motor untuk layanan taksi motor, melainkan hanya sebagai penghubung antara penumpang dan supir taksi motor.

### **2.2.5 Software as a Service**

*Software as a service* (SaaS) adalah sebuah lisensi perangkat lunak dan model bisnis dimana perangkat lunak sebagai berbasis layanan. SaaS semakin umum digunakan pada berbagai aplikasi bisnis, mulai dari aplikasi *office*, *messaging/chat*, pemrosesan pembayaran, perhitungan

akuntansi, *Customer Relationship Management* (CRM), solusi basis data, hingga *gamification* [18]. SaaS dinilai berguna bagi pebisnis Teknologi Informasi karena berpotensi mengurangi biaya *outsources* untuk pembelian dan perawatan investasi perangkat lunak dan perangkat keras.

Beberapa contoh dari SaaS adalah [veritrans.com](http://veritrans.com), yaitu solusi agregasi pembayaran untuk situs belanja *online*, dimana pengembang situs belanja *online* menggunakan layanan [veritrans.com](http://veritrans.com) agar pengguna mendapatkan akses pembayaran dari berbagai bank seperti via transfer Anjungan Tunai Mandiri (ATM), *e-banking* dan *mobile banking*, bahkan kartu kredit.

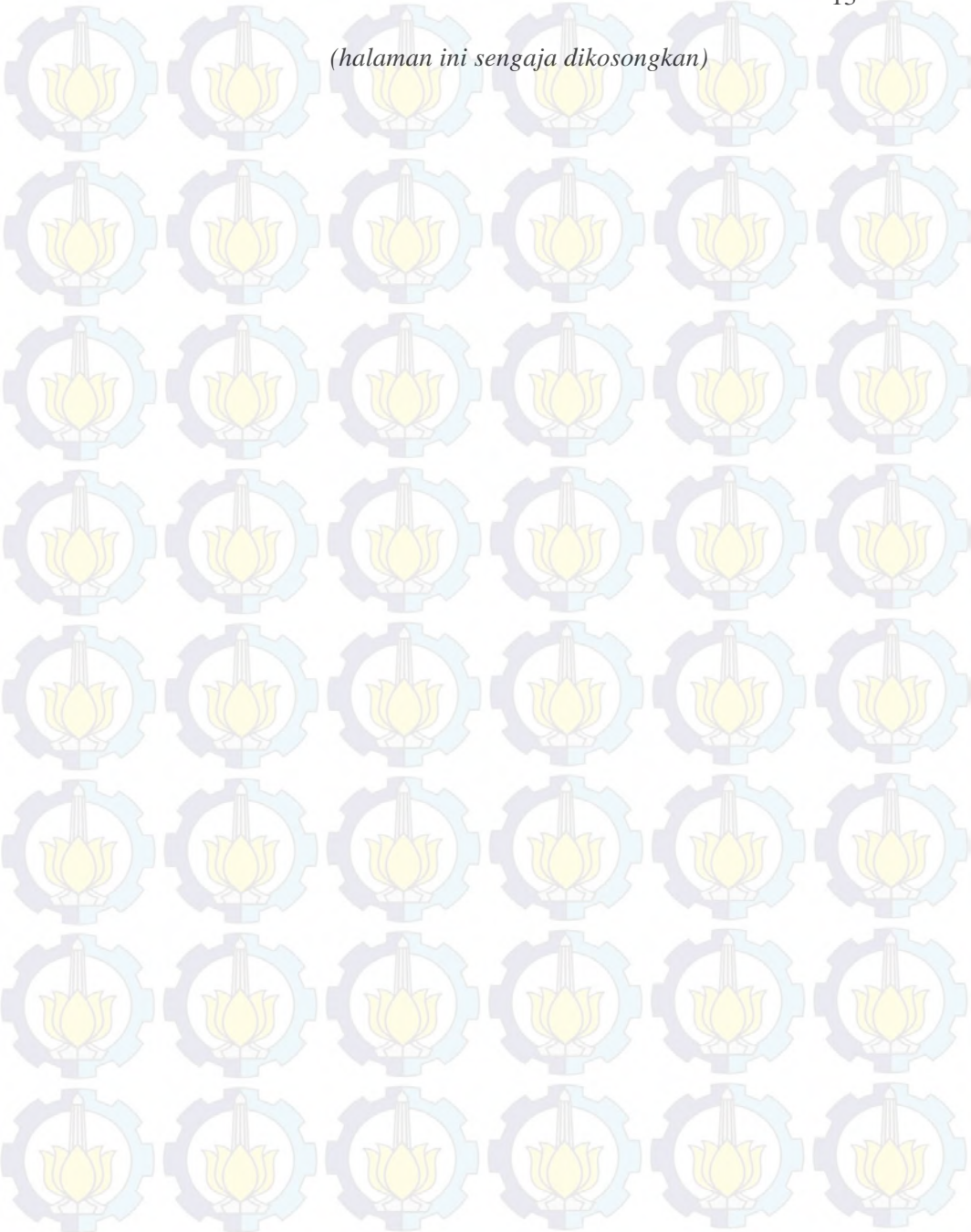
Aplikasi *Ojek Finder* menggunakan SaaS sebagai penyedia manajemen basis data yaitu [parse.com](http://parse.com). [Parse.com](http://Parse.com) menyediakan dashboard untuk mengelola basis data dalam server *cloud* [parse.com](http://parse.com). Aplikasi *Ojek Finder* menggunakan *Application Programming Interface* (API) yang disediakan [parse.com](http://parse.com) untuk mengakses basis data tersebut. [Parse.com](http://Parse.com) juga menyediakan layanan *push technology* untuk notifikasi *real-time* Aplikasi *Ojek Finder*.

### 2.2.6 Crowdsourcing

Crowdsourcing adalah gabungan dari kata *crowd* dan *outsourcing* yang berarti sebuah proses untuk mendapatkan layanan, ide, atau konten dengan mengumpulkan kontribusi dari kumpulan orang, biasanya menggunakan internet dari komunitas *online* [19].

Aplikasi *Ojek Finder* akan mengumpulkan supir taksi motor yang bersedia mengantarkan penumpang. Supir taksi motor akan memberikan tarif yang berbeda-beda kepada penumpang. Dengan mengumpulkan supir taksi motor penumpang dapat memilih harga yang paling sesuai menurut preferensi pribadi.

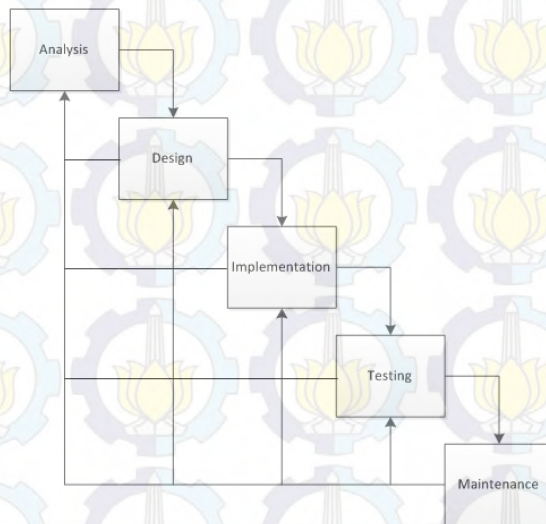
(halaman ini sengaja dikosongkan)





### BAB III METODOLOGI

Pada bagian ini akan dijelaskan rangkaian dari tahapan-tahapan yang dilakukan dalam pengerjaan Tugas Akhir ini. Rangkaian pengerjaan tugas akhir ini mengacu pada model pengembangan perangkat lunak *waterfall* sebagaimana digambarkan pada Gambar 3.1. Metode *waterfall* digunakan karena semua *requirement* sudah jelas di awal pengerjaan.



**Gambar 2 Model *Iterative Waterfall***

### 3.1 Tahapan Pelaksanaan Tugas Akhir



**Gambar 3 Tahap Pengerjaan Tugas Akhir**

#### 3.1.1 Studi Literatur

Studi literatur yang dilakukan antara lain pembelajaran dan pemahaman literatur mengenai konsep layanan transportasi berbasis *crowdsourcing* dan *sharing economy*, pembuatan diagram UML, pengembangan Aplikasi Android, dan dokumentasi dari parse.com.

#### 3.1.2 Analisis Kebutuhan Perangkat Lunak

Pada tahap ini dilakukan analisa *Functional* dan *Non-Functional Requirement* (FR & non-FR) pada Aplikasi Ojek Finder. Proses pencarian FR & non-FR dilakukan dengan *brainstorming*, riset terhadap aplikasi lain yang serupa tapi tidak sama, melakukan wawancara non-formal kepada



masyarakat yang biasa menggunakan layanan taksi motor dan masyarakat yang berprofesi sebagai supir taksi motor.

### 3.1.3 Desain Aplikasi

Dalam tahap ini akan dibuat berbagai diagram UML untuk membantu pengkodean aplikasi pada tahapan selanjutnya. Diagram seperti *use case* dibuat berdasarkan kebutuhan (FR & non-FR) perangkat lunak yang sudah didefinisikan pada tahap sebelumnya lalu dilanjutkan dengan pembuatan *activity diagram* dan *sequence diagram*. Pada tahap ini juga dibuat rancangan dari *database* sistem beserta relasinya. Setelah itu pada tahapan ini juga akan dibuat desain antar muka dari Aplikasi Ojek Finder serta arsitektur yang digunakan.

### 3.1.4 Pengkodean Aplikasi

Tahapan ini akan dimulai proses *coding* atau pengkodean dari aplikasi berdasarkan desain yang telah dibuat sebelumnya. Pada tugas akhir ini, bahasa pemrograman yang digunakan untuk *client* adalah *JAVA* dengan *Android SDK* sebagai *framework*. *SQLite* sebagai implementasi basis data. Untuk *Kembali-end* Ojek Finder menggunakan *back-end* dari *parse.com*. Sementara itu *Source Code Management* menggunakan *Git* dan *GitHub* sebagai *repository*.

### 3.1.5 Pengujian Aplikasi

Setelah pengkodean aplikasi telah selesai. Pengujian aplikasi dilakukan untuk mengecek apakah aplikasi sudah berjalan dengan baik atau belum sesuai dengan FR dan non-FR yang diwujudkan dalam *Requirement Traceability Matrix*. Untuk menguji kelayakan dari setiap komponen maka pada tahap ini juga akan dilakukan pengujian *unit testing*. Apabila masih ada beberapa komponen yang belum sempurna atau tidak memenuhi kriteria pengujian maka perbaikan akan dilakukan sampai perangkat lunak dapat berfungsi dengan baik.

### 3.1.6 Penyusunan Laporan Tugas Akhir

Apabila Aplikasi sudah lulus pengujian tahapan selanjutnya adalah membuat dokumentasi proses pengembangan perangkat lunak dari mulai identifikasi kebutuhan sampai dengan pengujian aplikasi. Dokumentasi tersebut akan diproyeksikan ke dalam bentuk laporan tugas akhir.



## BAB IV

### ANALISIS DAN DESAIN SISTEM “OJEK FINDER”

Pada bab ini, akan dijelaskan mengenai proses pembuatan “Ojek Finder” dari analisis kebutuhan sistem hingga perancangan desain sistem

#### 4.1 Benchmarking

Peneliti melakukan perbandingan terhadap aplikasi *Ridesharing* yang serupa dengan “Ojek Finder” seperti *Gojek*, *GrabTaxi*, dan *Uber* untuk mendapatkan *requirement specification*.

##### 4.1.1 Gojek

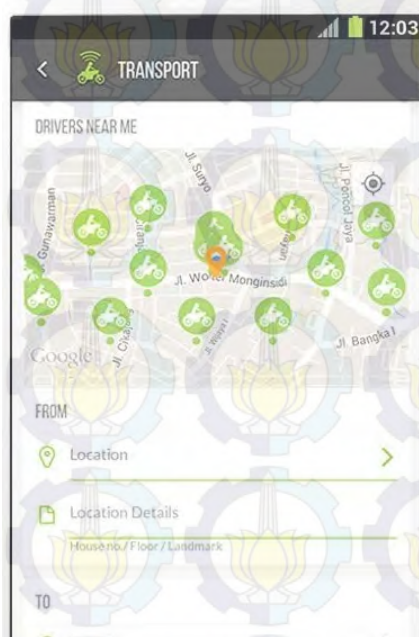
Gojek merupakan aplikasi pemesanan taksi motor online yang didirikan oleh Nadiem Makarim sejak tahun 2011. Gojek tidak hanya mengandallkan satu fitur pemesanan taksi motor online, namun juga fitur seperti antar makanan, pengiriman barang, dan mengirimkan barang belanjaan. Gojek memiliki istilah sendiri untuk fitur-fitur tersebut, seperti *Go-Food*, *Go-Mart*, dan seterusnya. Hingga tahun 2015 fitur pemesanan tersebut berkembang hingga integrasi dengan bus kota dan pelayanan jasa seperti *massage* dan tukang cukur rambut.



**Gambar 4 Menu Utama Gojek**

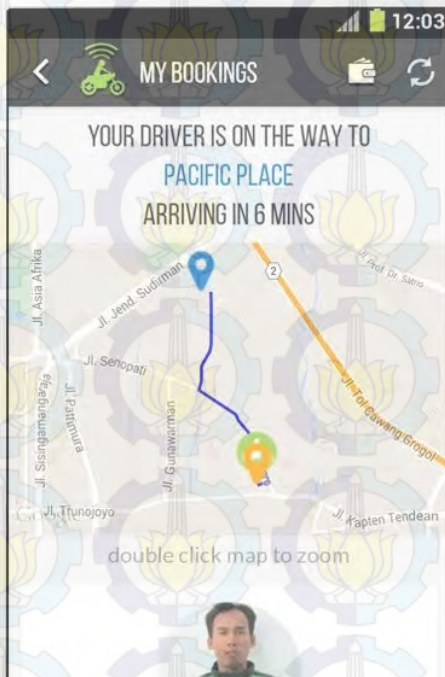
Desain tampilan awal Gojek diawali dengan kumpulan menu yang berbeda untuk setiap fitur. Gojek memiliki aplikasi yang berbeda untuk supir taksi motor. Sehingga tampilan untuk supir taksi motor dan calon penumpang berbeda.





**Gambar 5 Menu Pengemudi Di Sekitar Pengguna**

Hampir dalam setiap fitur gojek tersedia daftar spasial supir motor di sekitar kita. Dengan demikian pengguna bisa menjadikan informasi tersebut sebagai pertimbangan dalam setiap transaksi, bila dalam radius 10 km tidak ditemukan supir taksi motor di sekitar pengguna maka kemungkinan mendapatkan supir taksi motor menjadi rendah.

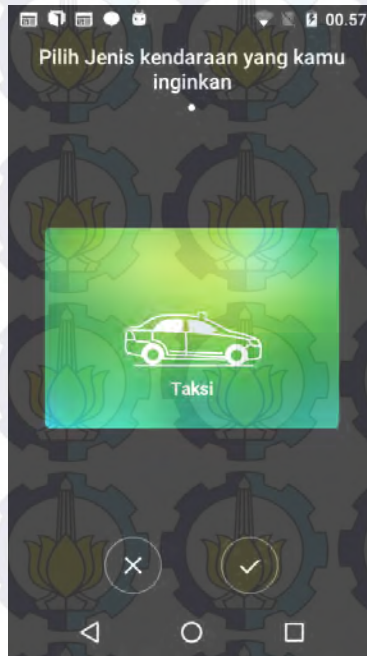


**Gambar 6 Tampilan Bila Terjadi Persetujuan Antara Penumpang dan Supir Gojek**

Setelah terjadi persetujuan dari pengguna dan supir taksi motor, Gojek akan memberikan informasi mengenai informasi ringkasan transaksi, seperti siapa supir taksi motor dan jarak antara supir dan pengguna.

#### **4.1.2 GrabTaxi**

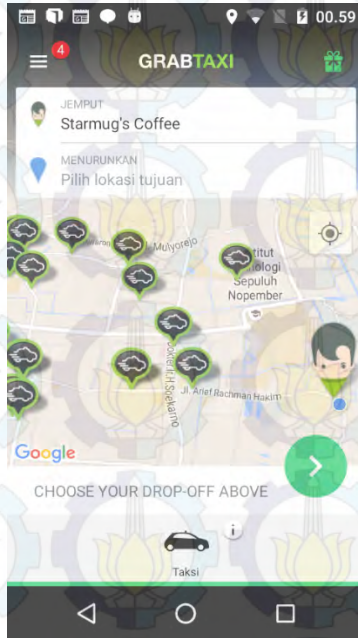
GrabTaxi merupakan aplikasi pemesanan taksi motor dan taksi mobil secara *online*. GrabTaxi merupakan fitur khusus GrabTaxi untuk pemesanan taksi motor *online*. GrabTaxi atau GrabTaxi didirikan di Malaysia oleh Anthony Tan dan Tan Hooi Ling sejak tahun 2011.



**Gambar 7 Pilihan Kendaraan Dalam GrabTaxi**

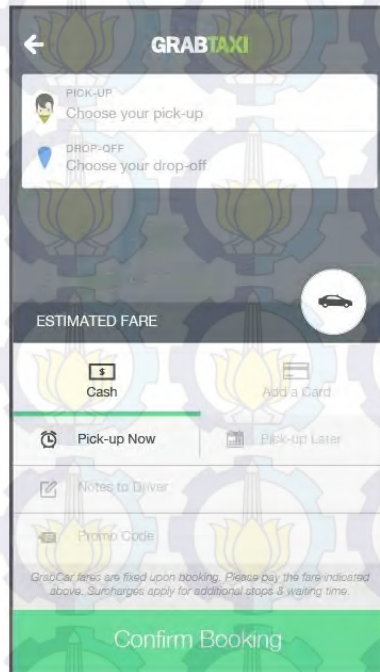
Dari segi tampilan, GrabTaxi langsung menampilkan *map-picker*, dimana pengguna langsung memilih lokasi penjemputan dan lokasi tujuan. Dan selanjutnya terdapat menu untuk memilih kendaraan yang digunakan oleh supir, sepeda motor atau mobil.





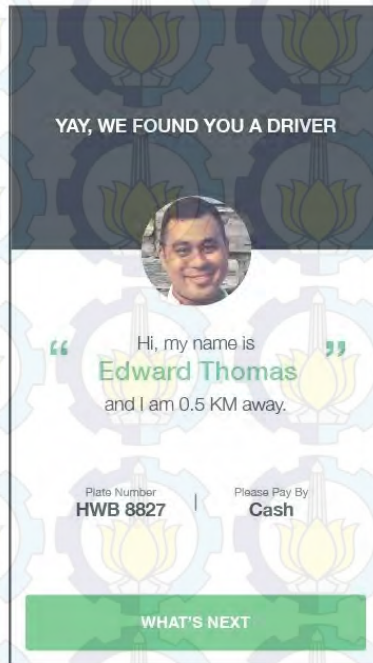
**Gambar 8 Tampilan Pengemudi Di Sekitar Pengguna**

GrabTaxi juga memiliki fitur untuk mengetahui keberadaan supir di sekitar pengguna. Dengan *icon* mobil untuk taksi mobil dan *icon* motor untuk taksi motor.



**Gambar 9 Tampilan Pemilihan Lokasi Keberangkatan dan Tujuan**

Setelah pengguna memberikan permintaan tumpangan, GrabTaxi akan mencari supir taksi motor atau mobil yang bersedia mengantarkan penumpang. GrabTaxi juga memiliki kolom untuk mengisi informasi tambahan dan kolom tips untuk masukan tambahan kepada supir.



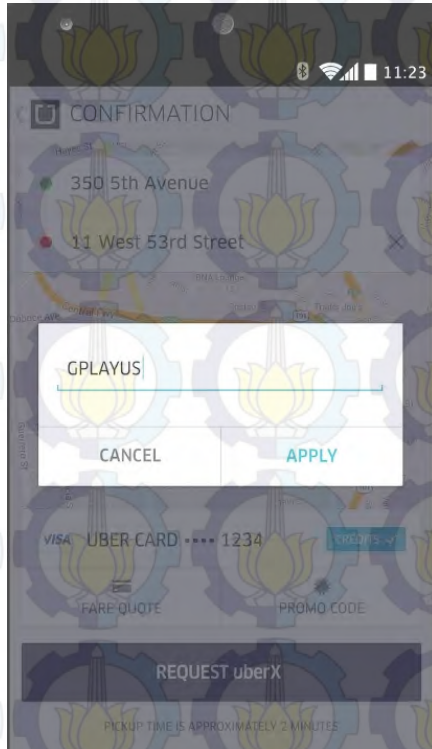
**Gambar 10 Tampilan Penjemputan Penumpang Oleh Supir**

Bila terjadi persetujuan antara supir dan penumpang maka akan muncul tampilan mengenai ringkasan transaksi layanan transportasi tersebut, seperti identitas supir nomor kendaraan dan metode pembayaran.

#### **4.1.3 Uber**

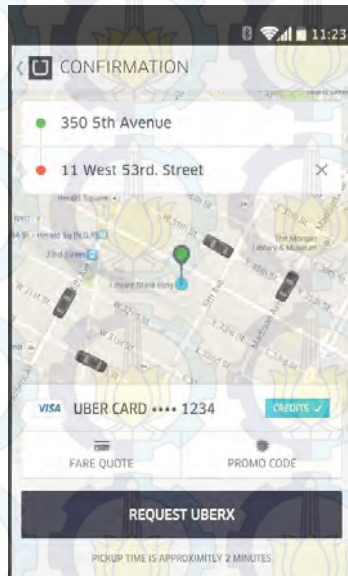
Uber merupakan aplikasi pemesanan transportasi mobil yang cukup populer di Amerika dan mulai tersedia di Indonesia sejak 2014. Uber didirikan di San Fransisco oleh Travis Kalanick dan Garrett Clamp pada tahun 2009. Uber memiliki visi untuk memberdayakan mobil yang dimiliki masyarakat sebagai alat transportasi.





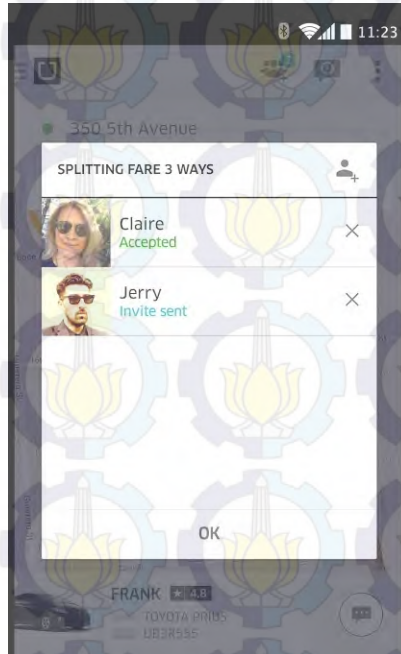
**Gambar 11 Penambahan Informasi Detil Transaksi**

Yang membedakan Uber dan Gojek dan GrabTaxi adalah metode pembayarannya. Dimana pengguna Uber menggunakan kartu kredit sebagai pembayaran. Kartu kredit dianggap fitur keamanan yang cukup membantu meningkatkan kepercayaan dan kredibilitas antar pengguna.



**Gambar 12 Konfirmasi Pemesanan Taksi**

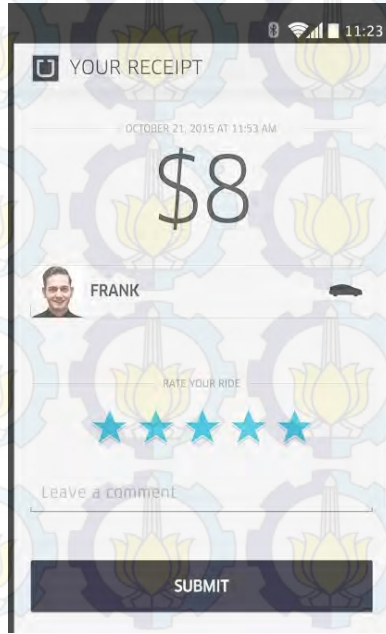
Fitur lain yang dimiliki oleh Uber adalah pembagian tarif transportasi dengan pengguna lain, atau *patungan*. Dengan fitur ini pengguna bisa mendapatkan transportasi yang lebih murah dengan berbagi tarif dengan pengguna lain.



**Gambar 13 Fitur Pembagian Ongkos Kirim**

Setelah terjadi persetujuan antara pengguna dan supir taksi motor Uber akan menampilkan rangkuman informasi transaksi, seperti mobil yang digunakan supir dan nomor kendaraannya. Serta *rating* dari supir tersebut.





**Gambar 14 Pemberian Rating dan Review**

Pengguna bisa memberikan *rating* dan *review* kepada supir taksi mobil bila transaksi sudah selesai. Dengan demikian pengguna bisa menggunakan informasi *rating* dan *review* supir tersebut sebagai pertimbangan dalam memilih supir taksi mobil.

#### **4.1.4 Requirement Specification**

Kumpulan spesifikasi fitur dari *benchmark* Gojek, GrabTaxi, dan Uber bisa kita uraikan dalam Tabel 2.

**Tabel 2 Perbandingan Fitur Gojek, GrabTaxi, dan Uber**

<b>Spesifikasi</b>	<b>Gojek</b>	<b>GrabTaxi</b>	<b>Uber</b>
Login Menggunakan Email	Ya	Ya	Ya

Aplikasi Khusus Untuk Admin	Ya	Ya	Ya
Akomodasi Transportasi Motor	Ya	Ya	Tidak
Akomodasi Transportasi Mobil	Tidak	Ya	Ya
Pantau Posisi Supir	Ya	Ya	Ya
Informasi Posisi Pengguna Lain	Ya	Ya	Ya
Pemilihan Lokasi Penjemputan dan Tujuan	Ya	Ya	Ya
Memberi <i>Review</i> dan <i>Rating</i> Kepada Supir	Ya	Ya	Ya
Mengatur Profil Pengguna	Ya	Ya	Ya
Pembatalan Transaksi	Ya	Ya	Ya
Pemberian Tips Kepada Supir	Tidak	Ya	Tidak
<i>Material Design</i>	Tidak	Ya	Ya

## 4.2 Analisis dan Desain “Ojek Finder”

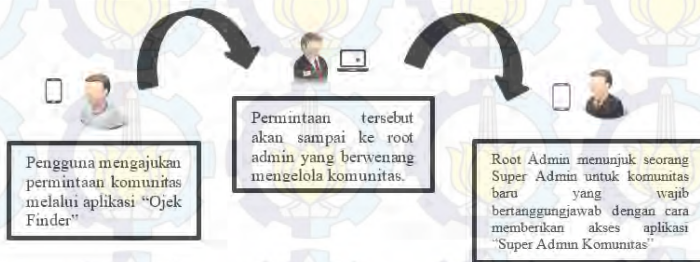
Pada bagian ini akan dijelaskan mengenai analisis dan desain dalam pengerjaan Ojek Finder, analisis dimulai dari deskripsi aplikasi, proses bisnis, kebutuhan fungsional dan non-fungsional, aktor-aktor dalam aplikasi dan *Use Case*. Sementara desain dimulai dari *Sequence Diagram*, *Class Diagram*, *Entity* dan *Relationship Diagram*.

### 4.2.1 Deskripsi Aplikasi “Ojek Finder”

Ojek Finder adalah aplikasi pemesanan taksi motor secara *online* berbasis komunitas. Pengguna Ojek Finder dapat mendaftarkan diri dan nantinya dapat bertindak sebagai penumpang atau supir taksi motor. Supir taksi motor diharuskan tergabung dalam komunitas, bila supir taksi motor belum tergabung maka calon penumpang akan melihat status supir taksi motor tersebut sebagai belum terverifikasi. Komunitas adalah sebuah kelompok dimana setiap anggotanya memiliki satu kesamaan niat, baik itu kesamaan kepercayaan, lokasi, ataupun status. Dengan Ojek Finder pengguna dapat memesan taksi motor kepada pengguna lain tanpa harus terikat kontrak kerja terhadap Ojek Finder.

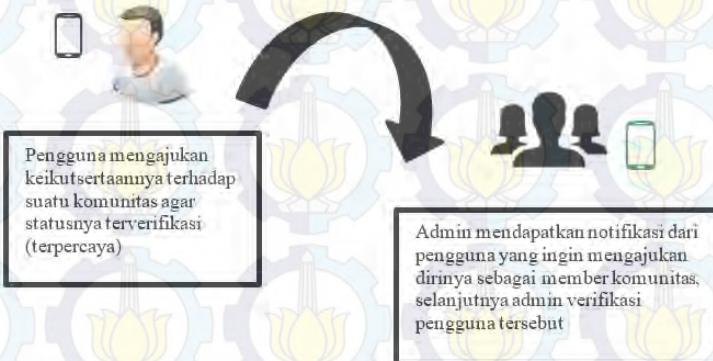
#### 4.2.2 Proses Bisnis Ojek Finder

Dalam Gambar 15 dijelaskan pengajuan suatu komunitas baru beserta seseorang *super admin* yang bertanggung jawab atas admin-admin komunitas yaitu orang-orang yang bertugas memverifikasi anggota komunitas.



**Gambar 15 Pengajuan Komunitas Baru**

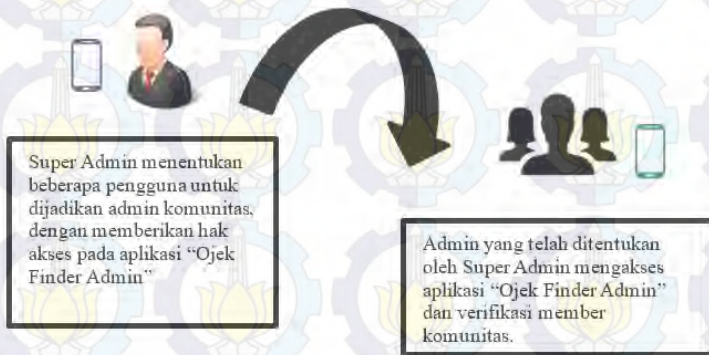
Dalam Gambar 16 dijelaskan bagaimana pengguna yang ingin bergabung dengan komunitas, terutama pengguna yang menjadi supir taksi motor, agar terverifikasi sebagai supir taksi motor terhadap suatu komunitas.



**Gambar 16 Pengajuan Verifikasi Pengguna**

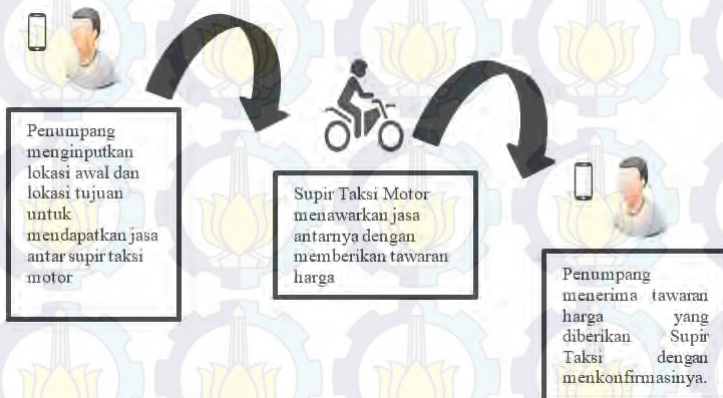


Dalam Gambar 17 dijelaskan cara pendaftaran admin komunitas dan verifikasi member komunitas.



**Gambar 17 Verifikasi Pengguna Oleh Admin**

Dalam Gambar 18 dijelaskan bagaimana pengguna menggunakan Ojek Finder untuk mencari layanan transportasi taksi motor dan menjadi supir taksi motor.



**Gambar 18 Permintaan Tumpangan Taksi Motor**

### 4.2.3 Kebutuhan Fungsional Sistem

Berdasarkan Tabel 3 didapatkan kebutuhan fungsional Ojek Finder yang sudah disesuaikan dengan latar belakang Ojek Finder. Kebutuhan fungsional sistem ini tertera pada Tabel 3.

**Tabel 3 Tabel Kebutuhan Fungsional**

ID	Kebutuhan Fungsional
FR-01	Penumpang dapat mengajukan pemesanan taksi motor
FR-02	Penumpang dapat melakukan persetujuan terhadap penawaran taksi motor
FR-03	Supir taksi motor dapat menawarkan tumpangan kepada calon penumpang
FR-04	Pengguna dapat melihat profil pengguna lain
FR-05	Pengguna dapat membatalkan transaksi
FR-06	Pengguna dapat memberi ulasan dan penilaian terhadap pengguna lain.
FR-07	Pengguna dapat melihat lokasi pengguna lain dalam peta
FR-08	Pengguna dapat mengelola profil pengguna
FR-09	Pengguna dapat mengelola akun
FR-10	Pengguna dapat mengajukan komunitas baru
FR-11	Admin Komunitas dapat mengelola status akun anggota komunitas
FR-12	Admin Komunitas dapat mengelola profil komunitas
FR-13	Admin Komunitas dapat melihat profil anggota komunitas
FR-14	Admin Komunitas dapat mengelola akun
FR-15	Super Admin Komunitas dapat mengelola Admin Komunitas
FR-16	Root Admin dapat mengelola Super Admin Komunitas
FR-17	Root Admin dapat menerima pengajuan komunitas baru
FR-18	Root Admin dapat mengelola status komunitas



#### 4.2.4 Kebutuhan non-fungsional sistem

Dari hasil analisis kebutuhan yang telah dilakukan, didapatkan kebutuhan fungsional berikut sebagaimana terdapat pada Tabel 4.

**Tabel 4 Tabel Kebutuhan Non-Fungsional**

ID	Kebutuhan Non-Fungsional	Jenis
NFR-01	Sistem dapat diakses 24 jam sehari, 7 hari seminggu	<i>Accessibility</i>
NFR-02	Sistem hanya bisa diakses menggunakan sistem operasi Android dengan versi minimal 4.0 (Ice Cream Sandwich)	<i>Accessibility</i>
NFR-03	Sistem harus membutuhkan koneksi internet minimal tingkat koneksi <i>Edge</i>	<i>Accessibility</i>
NFR-04	Sistem harus berhasil dalam menyimpan semua transaksi, baik profil dan ulasan pengguna, lokasi pengguna maupun permintaan dan penawaran taksi motor.	<i>Reliability</i>
NFR-05	Sistem menggunakan bahasa Indonesia dan bahasa Inggris	<i>Communication</i>
NFR-06	Sistem dibuat tanpa menyalahi ketentuan hukum Hak Kekayaan Intelektual	<i>Legal</i>

#### 4.2.5 Aktor-Aktor Dalam Sistem

Aktor-aktor yang terlibat pada sistem serta deskripsinya dijelaskan pada Tabel 5.

**Tabel 5 Tabel Aktor**

Aktor	Deskripsi
Pengguna	Semua orang terdaftar dalam aplikasi dan nantinya bisa berperan sebagai supir taksi motor atau penumpang.
Supir taksi motor	Pengguna yang berperan sebagai supir taksi motor yang menawarkan transaksi transportasi. Untuk menjadi Supir Taksi Motor pengguna harus terdaftar menjadi anggota komunitas.



Aktor	Deskripsi
Penumpang	Pengguna yang berperan sebagai penumpang taksi motor yang mencari supir taksi motor.
Admin Komunitas	Orang yang melakukan verifikasi dan validasi pengguna dalam komunitas.
Super Admin Komunitas	Bertanggung jawab untuk menambah, mengubah, mendeaktifasi, mengaktifasi Admin Komunitas.
Root	Bertanggung jawab untuk menambah, mengubah, mendeaktifasi, mengaktifasi Super Admin Komunitas dan Komunitas.

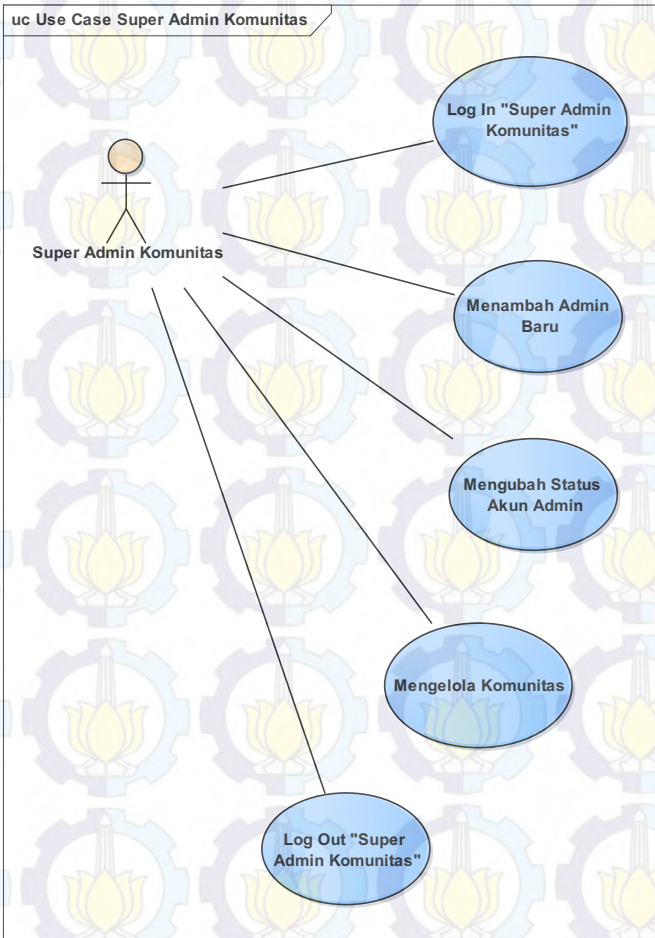
#### 4.2.6 Use Case

Berdasarkan analisis kebutuhan yang dilakukan, fungsionalitas dalam sistem direpresentasikan ke dalam beberapa *use case*. Daftar *use case* beserta *mappingnya* ke kebutuhan fungsional dapat dilihat pada tabel.

**Tabel 6 Tabel Pemetaan FR dan UC**

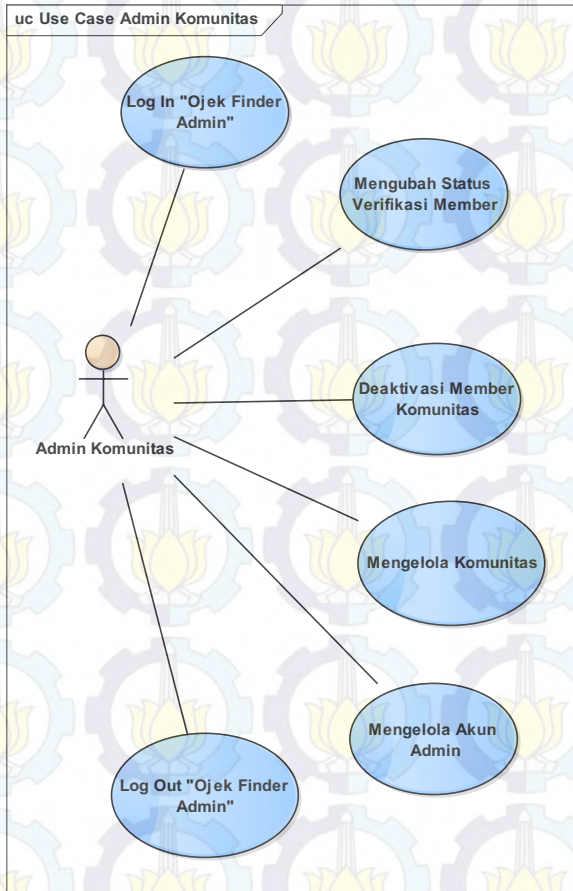
FR ID	UC ID	Use Case
FR-09	UC01	Sign Up “Ojek Finder”
FR-09	UC02	Log In “Ojek Finder”
FR-08	UC03	Mengelola profil
FR-09	UC04	Deaktifasi akun
FR-04	UC05	Melihat profil pengguna lain
FR-07	UC06	Melihat lokasi seorang pengguna
FR-07	UC07	Melihat lokasi seluruh pengguna di dekatnya
FR-10	UC08	Mengajukan permintaan komunitas baru
FR-01	UC09	Membatalkan transaksi
FR-06	UC10	Memberikan rating dan review
FR-09	UC11	Log Out “Ojek Finder”
FR-03	UC12	Memberikan tawaran tumpangan taksi motor

FR ID	UC ID	Use Case
FR-01	UC13	Membuat pemesanan taksi motor baru
FR-02	UC14	Menyetujui tawaran taksi motor
FR-11	UC15	Log In “Super Admin Komunitas”
FR-14	UC16	Mengubah Status Akun Admin
FR-15	UC17	Mengelola Komunitas
FR-15	UC18	Menambah Admin Baru
FR-15	UC19	Log Out “Super Admin Komunitas”
FR-14	UC20	Log In “Ojek Finder Admin”
FR-11	UC21	Mengubah status verifikasi member komunitas
FR-11	UC22	Deaktifasi member komunitas
FR-14	UC23	Mengelola akun admin
FR-12	UC24	Mengelola Komunitas
FR-14	UC25	Log Out “Ojek Finder Admin”
FR-16	UC26	Log In Root Admin
FR-18	UC27	Mengelola Status Komunitas
FR-16	UC28	Mengelola Super Admin Komunitas
FR-17	UC29	Menerima Pengajuan Komunitas Baru
FR-16	UC30	Log Out Root Admin

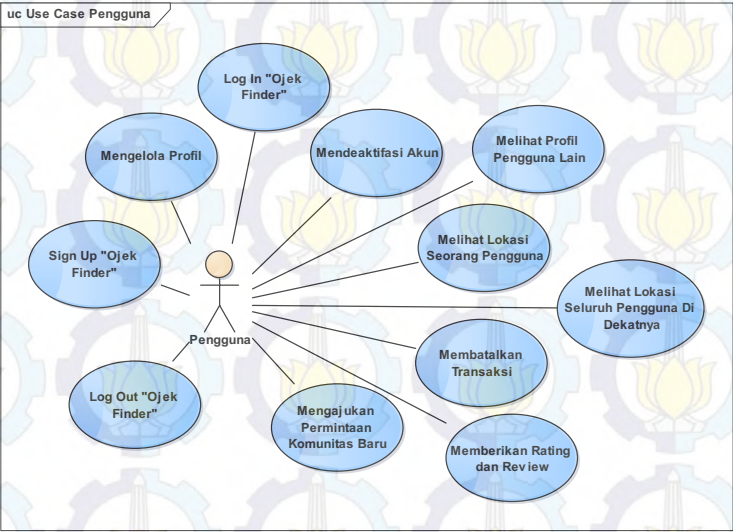


**Gambar 19 Use Case Super Admin Komunitas**

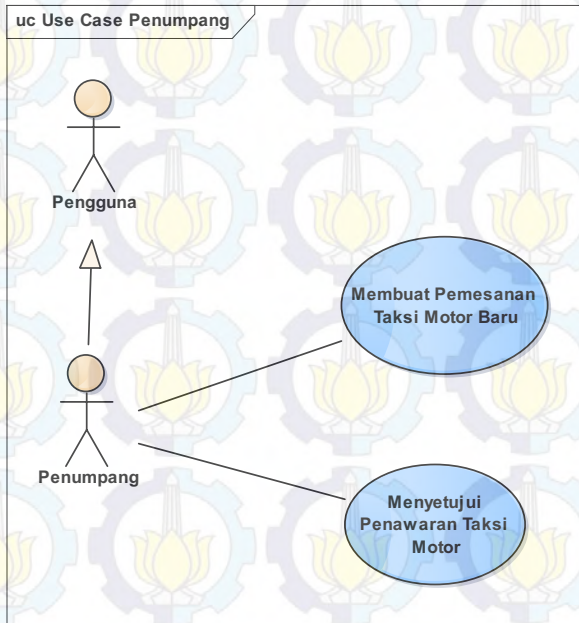




Gambar 20 Use Case Admin Komunitas

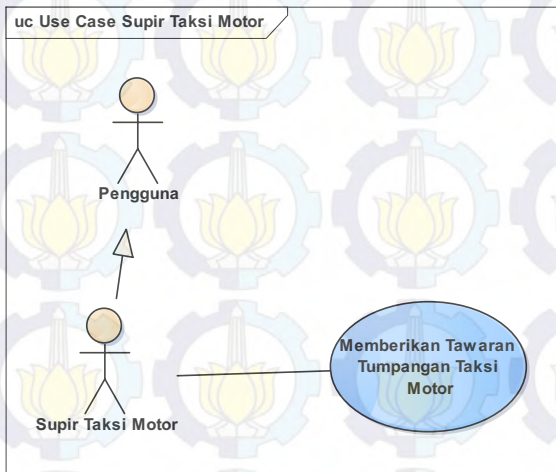


Gambar 21 Use Case Pengguna

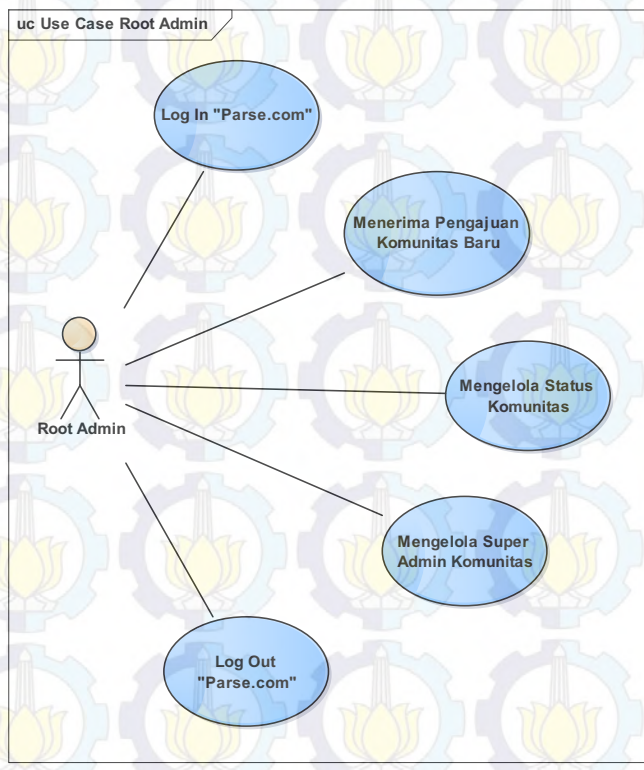


**Gambar 22 Use Case Penumpang**





Gambar 23 Use Case Supir Taksir Motor



Gambar 24 Use Case Root Admin

4.2.7 Use Case Description

Tabel 7 UCD Pendaftaran Pengguna

UC01: Pendaftaran Pengguna (Sign Up)	
Ringkasan:	Use Case ini dimaksudkan bahwa pengguna yang baru menggunakan aplikasi "Ojek Finder" dan akan melakukan <i>Sign Up</i>
Aktor:	Pengguna
Prioritas:	Penting ( <i>essential</i> )
Frekuensi:	Dilakukan satu kali

UC01: Pendaftaran Pengguna (Sign Up)	
Kondisi:	Pengguna telah menginstall aplikasi pada mobile Android
Skenario Sukses Utama:	<ol style="list-style-type: none"> <li>1. Pengguna membuka aplikasi "Ojek Driver" dan memilih tombol "Sign Up"</li> <li>2. Pengguna Memasukkan E-mail dan Password, lalu menginputkan password lagi untuk mengecek apakah terdapat <i>human error</i> pada saat memasukkan password yang pertama</li> <li>3. Pengguna menekan tombol <i>Sign Up</i>, lalu pengguna masuk ke halaman pertama aplikasi.</li> </ol>
Skenario Alternatif:	<ol style="list-style-type: none"> <li>1. Apabila pengguna tidak mengisi kolom email atau password maka system menampilkan pesan <i>error</i>.</li> <li>2. Apabila pengguna tidak memasukkan password again sesuai dengan kolom password sebelumnya maka system menampilkan pesan <i>error</i>.</li> </ol>

Tabel 8 UCD Log In Aplikasi Ojek Finder

UC02: Log In Aplikasi Ojek Finder	
Ringkasan:	Use Case ini dimaksudkan agar pengguna dapat masuk ke dalam aplikasi "Ojek Finder" sesuai akun yang telah dibuat sebelumnya.
Aktor:	Pengguna
Prioritas:	Penting (essential)
Frekuensi:	Dapat dilakukan lebih dari satu kali
Kondisi:	Pengguna telah melakukan sign up pada aplikasi "Ojek Finder" sebelumnya



UC02: Log In Aplikasi Ojek Finder	
Skenario Sukses Utama:	<ol style="list-style-type: none"><li>1. Pengguna masuk ke halaman persembahan dari aplikasi lalu menekan tombol Log In</li><li>2. Pengguna mengetikkan email dan password pada masing-masing kolom.</li><li>3. Pengguna menekan tombol Log In</li><li>4. Pengguna berhasil masuk ke halaman akunnya.</li></ol>
Skenario Alternatif:	<ol style="list-style-type: none"><li>1. Pengguna mengetikkan email yang tidak terdaftar sebagai member, maka system menampilkan pesan error.</li><li>2. Pengguna mengetikkan password yang tidak sesuai dengan e-mail pada akun yang terdaftar sebagai member, maka system menampilkan pesan error.</li></ol>

**Tabel 9 UCD Mengelola Profil**

UC03: Mengelola Profil	
Ringkasan:	Use Case ini dimaksudkan bahwa pengguna dapat merubah profil terkait nama, foto, dan nomor HP
Aktor:	Pengguna
Prioritas:	Penting (essential)
Frekuensi:	Bisa lebih dari satu kali
Kondisi:	Pengguna telah sign up atau log in aplikasi Ojek Finder
Skenario Sukses Utama:	<ol style="list-style-type: none"> <li>1. Pengguna menggeser halaman utama ke kanan, muncul tampilan Profil</li> <li>2. Pengguna menekan foto profil lalu terintegrasi ke photo library pada handphone, pengguna memilih foto dan mengukur (crop) sesuai dengan yang diinginkan, lalu klik tombol pilih.</li> <li>3. Pengguna menekan ikon edit pada nama profil, maka pengguna dapat mengedit nama profil.</li> <li>4. Pengguna menekan ikon edit pada informasi nomor HP, maka pengguna dapat mengedit nomor HP yang sesuai.</li> <li>5. Pengguna menekan tombol “Simpan” setelah selesai mengedit. Maka profil diperbarui.</li> </ol>
Skenario Alternatif:	<ol style="list-style-type: none"> <li>1. Pengguna memasukkan angka ke dalam kolom nama, maka system akan menampilkan pesan error saat pengguna telah menekan tombol Simpan.</li> <li>2. Pengguna memasukkan huruf ke</li> </ol>

UC03: Mengelola Profil	
	<p>dalam kolom nomor HP, maka system akan menampilkan pesan error saat pengguna telah menekan tombol Simpan.</p> <p>3. Pengguna menekan tombol “Kembali” sebelum menekan tombol “Simpan”. Maka profil tidak diperbarui.</p>

**Tabel 10 UCD Deaktivasi Akun**

UC04: Deaktivasi Akun	
Ringkasan:	Use Case ini dimaksudkan agar pengguna dapat menonaktifkan keikutsertaannya sebagai member pada salah satu komunitas di aplikasi “Ojek Finder”.
Aktor:	Pengguna
Prioritas:	Pilihan (Optional)
Frekuensi:	Dapat dilakukan lebih dari satu kali
Kondisi:	Pengguna telah melakukan log in pada aplikasi “Ojek Finder”.
Skenario Sukses Utama:	<ol style="list-style-type: none"> <li>1. Pengguna menggeser halaman utama ke kanan, muncul tampilan Profil</li> <li>2. Pengguna menekan tombol “Deactivate Account”, maka akun yang telah dibuat sebelumnya menjadi nonaktif.</li> </ol>
Skenario Alternatif:	-



**Tabel 11 UCD Melihat Profil Pengguna Lain**

UC05: Melihat Profil Pengguna Lain	
Ringkasan:	Use Case ini dimaksudkan agar pengguna dapat melihat profil pengguna lain pada aplikasi “Ojek Finder” pada bar Penumpang.
Aktor:	Pengguna
Prioritas:	Penting (essential)
Frekuensi:	Dapat dilakukan lebih dari satu kali
Kondisi:	Pengguna telah melakukan log in pada aplikasi “Ojek Finder”.
Skenario Sukses Utama:	<ol style="list-style-type: none"> <li>1. Aktor masuk ke bar “Penumpang” dan dapat melihat daftar penumpang yang belum mendapatkan konfirmasi dengan pihak ojek.</li> <li>2. Pengguna dapat melihat rincian profil penumpang dengan menekan salah satu daftar penumpang. Rincian yang dapat dilihat meliputi: Nama penumpang serta foto profilnya, lokasi awal, lokasi tujuan dan detail tambahan.</li> </ol>
Skenario Alternatif:	-

**Tabel 12 UCD Melihat Lokasi Seorang Pengguna**

UC06: Melihat Lokasi Seorang Pengguna	
Ringkasan:	Use Case ini dimaksudkan agar pengguna dapat melihat lokasi keberadaannya.
Aktor:	Pengguna
Prioritas:	Pilihan (optional)
Frekuensi:	Dapat dilakukan lebih dari satu kali
Kondisi:	Pengguna telah melakukan log in pada aplikasi “Ojek Finder”.

UC06: Melihat Lokasi Seorang Pengguna	
Skenario Sukses Utama:	<ol style="list-style-type: none"> <li>1. Pengguna menekan konteks menu pengguna lain daftar pengguna, maka muncul tampilan peta lokasi pengguna.</li> <li>2. Pengguna mendapatkan titik lokasi keberadaannya.</li> </ol>
Skenario Alternatif:	<ol style="list-style-type: none"> <li>1. Pengguna gagal mendapatkan lokasi keberadaannya dikarenakan jaringan internet tidak memadai sehingga lokasi yang ditunjukkan tidak sesuai atau ada pada titik terakhir pengguna melihat lokasi.</li> </ol>

**Tabel 13 UCD Melihat Lokasi Seluruh Pengguna Terdekat**

UC07: Melihat Lokasi Seluruh Pengguna Terdekat	
Ringkasan:	Use Case ini dimaksudkan agar pengguna dapat melihat lokasi pengguna lain yang di dekatnya dengan jarak 5 km.
Aktor:	Pengguna
Prioritas:	Pilihan (optional)
Frekuensi:	Dapat dilakukan lebih dari satu kali
Kondisi:	Pengguna telah melakukan log in pada aplikasi “Ojek Finder”.
Skenario Sukses Utama:	<ol style="list-style-type: none"> <li>1. Pengguna menekan ikon maps pada ujung kanan atas aplikasi pada halaman utama.</li> <li>2. Pengguna mendapatkan titik lokasi pengguna lain yang berada di dekatnya.</li> </ol>
Skenario Alternatif:	<ol style="list-style-type: none"> <li>1. Pengguna gagal mendapatkan lokasi pengguna lain dikarenakan jaringan internet tidak memadai sehingga lokasi yang ditunjukkan</li> </ol>

UC07: Melihat Lokasi Seluruh Pengguna Terdekat	
	tidak sesuai atau ada pada titik terakhir pengguna melihat lokasi.

**Tabel 14 UCD mengajukan Permintaan Komunitas Baru**

UC08: Mengajukan Permintaan Komunitas Baru	
Ringkasan:	Use Case ini dimaksudkan agar pengguna dapat meminta kepada Root Admin untuk dibuat komunitas baru.
Aktor:	Pengguna
Prioritas:	Pilihan (optional)
Frekuensi:	Dilakukan satu kali
Kondisi:	Pengguna telah melakukan log in pada aplikasi “Ojek Finder”.
Skenario Sukses Utama:	<ol style="list-style-type: none"> <li>1. Pengguna menggeser halaman utama ke kanan, muncul tampilan Profil lalu menekan daftar tombol “Pengajuan Komunitas”</li> <li>2. Pengguna mengetikkan kontak yang bisa dihubungi berupa email, nomor HP, dan mengisi kolom komentar serta nama komunitas yang akan diajukan lalu menekan tombol “Simpan”, maka pengguna menunggu <i>feedKembali</i> dari Root Admin.</li> </ol>
Skenario Alternatif:	<ol style="list-style-type: none"> <li>1. Pengguna memasukkan huruf ke dalam kolom nomor HP, maka system akan menampilkan pesan error saat pengguna telah menekan tombol Simpan.</li> <li>2. Pengguna menekan tombol “Kembali” sebelum menekan tombol “Simpan”. Maka pengajuan tidak dikirim.</li> </ol>



Tabel 15 UCD Membatalkan Transaksi

UC09: Membatalkan Transaksi	
Ringkasan:	Use Case ini dimaksudkan agar pengguna dapat membatalkan transaksi yang dibuat sebelumnya.
Aktor:	Pengguna
Prioritas:	Pilihan (optional)
Frekuensi:	Dilakukan satu kali
Kondisi:	Pengguna telah melakukan log in pada aplikasi “Ojek Finder”, pengguna telah memberikan tawaran atau meminta tumpangan pada aplikasi.
Skenario Sukses Utama:	<ol style="list-style-type: none"> <li>1. Pengguna menekan bar “Setuju”, maka pengguna dapat melihat permintaan atau tawaran tumpangan yang telah dikonfirmasi.</li> <li>2. Pengguna menekan daftar konfirmasi yang telah dilakukan maka akan muncul window box pertanyaan “Apakah Anda akan Membatalkan Transaksi?”</li> <li>3. Pengguna menekan tombol “Ya”</li> <li>4. List konfirmasi yang telah dibatalkan terhapus.</li> </ol>
Skenario Alternatif:	<ol style="list-style-type: none"> <li>1. Pengguna menekan tombol “Tidak” pada window box, maka list konfirmasi yang akan dibatalkan tidak terhapus atau tetap berjalan.</li> </ol>

Tabel 16 UCD Memberikan Rating dan Review

UC10: Memberikan rating dan review	
Ringkasan:	Use Case ini dimaksudkan agar pengguna dapat memberikan timbal balik setelah mengantar atau diantar oleh pengguna lain dengan tujuan dapat dijadikan penilaian terhadap setiap pengguna oleh pengguna lain yang akan menggunakan jasanya.
Aktor:	Pengguna
Prioritas:	Penting (essential)
Frekuensi:	Dilakukan lebih dari satu kali
Kondisi:	Pengguna telah melakukan log in pada aplikasi “Ojek Finder”, pengguna telah memberikan tawaran atau meminta tumpangan pada aplikasi.
Skenario Sukses Utama:	<ol style="list-style-type: none"> <li>1. Pengguna menekan profil pengguna yang telah mengantar atau diantar.</li> <li>2. Pengguna menekan tombol Review dan Rating, maka muncul halaman dimana pengguna dapat mengetikkan hasil penilaian setelah diantar atau mengantarkan pengguna lain.</li> <li>3. Pengguna menekan tombol “Send”</li> </ol>
Skenario Alternatif:	<ol style="list-style-type: none"> <li>1. Pengguna tidak menekan tombol “Send” maka hasil review dan rating tidak tersampaikan pada pengguna lain yang akan diberi review dan rating.</li> </ol>

Tabel 17 UCD Log Out Ojek Finder

UC11: Log Out “Ojek Finder”	
Ringkasan:	Use Case ini dimaksudkan agar Admin Komunitas dapat keluar dari aplikasi “Ojek Finder”.
Aktor:	Pengguna
Prioritas:	Pilihan (Optional)
Frekuensi:	Bisa dilakukan lebih dari satu kali
Kondisi:	Pengguna telah log in aplikasi “Ojek Finder”
Skenario Sukses Utama:	<ol style="list-style-type: none"> <li>1. Pengguna menggeser layar ke kanan (navigation drawer) lalu menekan tombol “log out”, maka muncul halaman log in pada aplikasi “Ojek Finder”</li> </ol>
Skenario Alternatif:	-

Tabel 18 UCD Memberikan Tawaran Tumpangan Taksi Motor

UC12: Memberikan Tawaran Tumpangan Taksi Motor	
Ringkasan:	Use Case ini dimaksudkan agar Supir Taksi Motor dapat memberikan penawarannya terhadap penumpang yang ingin mendapatkan jasa antar dari lokasi awal hingga lokasi tujuan penumpang.
Aktor:	Supir Taksi Motor
Prioritas:	Penting (essential)
Frekuensi:	Dilakukan lebih dari satu kali
Kondisi:	Pengguna telah melakukan log in pada aplikasi “Ojek Finder”
Skenario Sukses Utama:	<ol style="list-style-type: none"> <li>1. Pengguna menekan profil penumpang pada bar penumpang yang akan diantar. Terdapat window box penawaran harga dan</li> </ol>



UC12: Memberikan Tawaran Tumpangan Taksi Motor	
	<p>detail informasi yang dapat diisi oleh Supir taksi motor.</p> <ol style="list-style-type: none"> <li>Supir Taksi Motor menekan tombol “Konfirmasi”</li> <li>Penawaran yang dilakukan akan masuk ke dalam bar “Setuju” setelah penumpang sepakat dengan penawaran yang diberikan Supir taksi yang akan dijelaskan pada UC13.</li> </ol>
Skenario Alternatif:	<ol style="list-style-type: none"> <li>Supir Taksi Motor tidak mengisi list harga untuk melakukan penawaran, maka system menampilkan pesan error.</li> <li>Supir Taksi Motor tidak mengisi detail tambahan untuk melakukan penawaran, maka penawaran tetap bisa terkirim karena detail tambahan tersebut bersifat optional.</li> </ol>

**Tabel 19 UCD Membuat Pemesanan taksi Motor Baru**

UC13: Membuat Pemesanan Taksi Motor Baru	
Ringkasan:	Use Case ini dimaksudkan agar Penumpang dapat membuat pemesanan taksi motor baru terhadap Supir taksi motor yang ingin mengantarkan dari lokasi awal hingga lokasi tujuan penumpang.
Aktor:	Penumpang
Prioritas:	Penting (essential)
Frekuensi:	Dilakukan lebih dari satu kali
Kondisi:	Pengguna telah melakukan log in pada aplikasi “Ojek Finder”
Skenario Sukses Utama:	<ol style="list-style-type: none"> <li>Penumpang menekan tombol</li> </ol>

## UC13: Membuat Pemesanan Taksi Motor Baru

floating action bar yang bersimbol (+) pada pojok kanan bawah layar smartphone, maka muncul halaman baru yang berisikan kolom lokasi awal, lokasi tujuan dan detail tambahan.

2. Penumpang menekan kolom lokasi awal yang akan muncul peta dari google, dan penumpang dapat menandai lokasi penjemputan untuk Supir taksi motor, lalu penumpang menekan tombol “Simpan” setelah menentukan lokasi yang sesuai.
3. Penumpang menekan kolom lokasi tujuan yang akan muncul peta dari google, dan penumpang dapat menandai lokasi tujuan pada saat menggunakan jasa Supir taksi motor, lalu penumpang menekan tombol “Simpan” setelah menentukan lokasi yang diinginkan.
4. Penumpang menekan kolom detail tambahan untuk mengetikkan keinginan atau ciri-ciri lokasi penjemputan.
5. Penumpang menekan tombol “Simpan” untuk melakukan pemesanan jasa Supir taksi motor.
6. Pemesanan yang dilakukan akan masuk ke dalam bar “Setuju” setelah Supir taksi sepakat dengan permintaan yang diberikan penumpang yang telah dijelaskan pada UC11.

## Skenario Alternatif:

1. Penumpang tidak menekan tombol “Simpan” pada saat menentukan

UC13: Membuat Pemesanan Taksi Motor Baru	
	<p>lokasi awal dan lokasi tujuan, maka tidak terjadi apa-apa pada halaman google maps tersebut.</p> <p>2. Penumpang tidak menekan tombol “Simpan” pada halaman pemesanan, maka pemesanan tidak dapat masuk ke daftar penumpang.</p>

**Tabel 20 UCD Menyetujui Penawaran Taksi Motor**

UC14: Menyetujui Penawaran Taksi Motor	
Ringkasan:	Use Case ini dimaksudkan agar Penumpang dapat membuat persetujuan pemesanan jasa taksi motor yang ingin mengantarkan dari lokasi awal hingga lokasi tujuan penumpang.
Aktor:	Penumpang
Prioritas:	Penting (essential)
Frekuensi:	Dilakukan lebih dari satu kali
Kondisi:	Pengguna telah melakukan log in pada aplikasi “Ojek Finder”
Skenario Sukses Utama:	<ol style="list-style-type: none"> <li>1. Pengguna menekan profil Supir taksi motor pada bar ojek yang telah melakukan penawaran. Terdapat window box penawaran konfirmasi setuju menggunakan jasa Supir taksi motor tersebut. Penumpang menekan tombol “Ya”</li> <li>2. Supir Taksi Motor menekan tombol “Konfirmasi”</li> <li>3. Penawaran yang dilakukan akan masuk ke dalam bar “Setuju” setelah penumpang sepakat dengan penawaran yang diberikan Supir taksi motor yang dipilih.</li> </ol>
Skenario Alternatif:	<ol style="list-style-type: none"> <li>1. Penumpang menekan tomol</li> </ol>



UC14: Menyetujui Penawaran Taksi Motor	
	“Tidak” pada window box penawaran konfirmasi, maka tawaran dari Supir taksi motor yang dipilih tidak terkonfirmasi atau tidak masuk dalam bar “Setuju”

**Tabel 21 UCD Log In Super Admin Komunitas**

UC15: Log In Super Admin Komunitas	
Ringkasan:	Use Case ini dimaksudkan agar Admin Komunitas dapat mengakses aplikasi “Super Admin Komunitas” yang bertujuan untuk mengelola admin pada sebuah komunitas.
Aktor:	Super Admin Komunitas
Prioritas:	Penting (essential)
Frekuensi:	Dilakukan satu kali
Kondisi:	Super Admin Komunitas telah menginstall aplikasi “Super Admin Komunitas”
Skenario Sukses Utama:	<ol style="list-style-type: none"> <li>1. Pengguna masuk ke halaman persembahan dari aplikasi lalu menekan tombol Log In</li> <li>2. Pengguna mengetikkan email dan password pada masing-masing kolom yang telah disetujui oleh Root Admin.</li> <li>3. Pengguna menekan tombol “Log In”</li> <li>4. Pengguna berhasil masuk ke halaman akunnya.</li> </ol>
Skenario Alternatif:	<ol style="list-style-type: none"> <li>1. Admin Komunitas mengetikkan email yang tidak disetujui Root Admin, maka system menampilkan pesan error.</li> <li>2. Pengguna mengetikkan password</li> </ol>

UC15: Log In Super Admin Komunitas	
	yang tidak disetujui Root Admin, maka system menampilkan pesan error.

**Tabel 22 UCD Mengubah Status Akun Admin**

UC16: Mengubah Status Akun Admin	
Ringkasan:	Use Case ini dimaksudkan agar Super Admin Komunitas dapat menentukan status pada admin yang ditunjuk pada aplikasi “Super Admin Komunitas”.
Aktor:	Super Admin Komunitas
Prioritas:	Penting (essential)
Frekuensi:	Bisa dilakukan lebih dari satu kali
Kondisi:	Super Admin Komunitas telah log in aplikasi “Super Admin Komunitas”
Skenario Sukses Utama:	<ol style="list-style-type: none"> <li>1. Super Admin Komunitas menekan profil admin yang terdapat pada list admin yang telah dibuat, maka muncul window box detail profil admin dan button “Active” or “Inactive”.</li> <li>2. Super Admin Komunitas menekan tombol “Activate”, maka admin yang dipilih statusnya aktif dan dapat mengakses aplikasi “Ojek Finder Admin”</li> </ol>
Skenario Alternatif:	<ol style="list-style-type: none"> <li>1. Super Admin Komunitas menekan tombol “Inactive”, maka admin yang dipilih statusnya tidak aktif dan tidak dapat mengakses aplikasi “Ojek Finder Admin”</li> </ol>

**Tabel 23 UCD Mengelola Komunitas**

UC17: Mengelola Komunitas	
Ringkasan:	Use Case ini dimaksudkan bahwa admin dapat merubah profil mengenai komunitas.
Aktor:	Super Admin Komunitas
Prioritas:	Pilihan (optional)
Frekuensi:	Bisa lebih dari satu kali
Kondisi:	Admin telah sign up atau log in aplikasi Super Admin Komunitas
Skenario Sukses Utama:	<ol style="list-style-type: none"> <li>1. Super Admin Komunitas menggeser halaman utama ke kanan, muncul tampilan Profil.</li> <li>2. Super Admin Komunitas menekan ikon edit pada nama komunitas, maka admin dapat mengedit nama komunitas.</li> <li>3. Super Admin Komunitas menekan tombol “Simpan” setelah selesai mengedit. Maka nama komunitas diperbarui.</li> </ol>
Skenario Alternatif:	<ol style="list-style-type: none"> <li>1. Super Admin Komunitas menekan tombol “Kembali” sebelum menekan tombol “Simpan”. Maka nama komunitas tidak diperbarui.</li> </ol>

**Tabel 24 UCD Menambah Admin Baru**

UC18: Menambah Admin Baru	
Ringkasan:	Use Case ini dimaksudkan agar Super Admin Komunitas dapat menambah admin baru melalui aplikasi “Super Admin Komunitas”.
Aktor:	Super Admin Komunitas
Prioritas:	Penting (essential)



UC18: Menambah Admin Baru	
Frekuensi:	Bisa dilakukan lebih dari satu kali
Kondisi:	Super Admin Komunitas telah log in aplikasi “Super Admin Komunitas”
Skenario Sukses Utama:	<ol style="list-style-type: none"> <li>1. Super Admin Komunitas menekan floating action bar pada pojok kanan bawah, maka muncul halaman pendaftaran admin baru dengan memasukkan email dan password admin baru.</li> <li>2. Super Admin Komunitas menekan tombol “Simpan”, maka email dan password yang telah disimpan dapat digunakan untuk hak akses pada aplikasi “Ojek Finder Admin”</li> </ol>
Skenario Alternatif:	<ol style="list-style-type: none"> <li>1. Super Admin Komunitas memasukkan email yang sama dengan email yang telah terdaftar maka akan muncul pesan error.</li> </ol>

**Tabel 25 UCD Log Out Aplikasi Super Admin Komunitas**

UC19: Log Out Aplikasi “Super Admin Komunitas”	
Ringkasan:	Use Case ini dimaksudkan agar Super Admin Komunitas dapat keluar dari aplikasi “Super Admin Komunitas”.
Aktor:	Super Admin Komunitas
Prioritas:	Penting (essential)
Frekuensi:	Bisa dilakukan lebih dari satu kali
Kondisi:	Super Admin Komunitas telah log in aplikasi “Super Admin Komunitas”
Skenario Sukses Utama:	<ol style="list-style-type: none"> <li>1. Super Admin Komunitas menekan tombol context menu pada pojok kanan atas, maka muncul action bar “log out”.</li> </ol>

UC19: Log Out Aplikasi “Super Admin Komunitas”	
	2. Super Admin Komunitas menekan action bar “log out”, maka muncul halaman log in pada aplikasi “Super Admin Komunitas”
Skenario Alternatif:	-

**Tabel 26 UCD Log In Ojek Finder Admin**

UC20: Log In Ojek Finder Admin	
Ringkasan:	Use Case ini dimaksudkan agar Admin Komunitas dapat mengakses aplikasi “Ojek Finder Admin” yang bertujuan untuk mengelola member pada sebuah komunitas.
Aktor:	Admin Komunitas
Prioritas:	Penting (essential)
Frekuensi:	Dilakukan satu kali
Kondisi:	Admin Komunitas telah menginstall aplikasi “Ojek Finder Admin”
Skenario Sukses Utama:	<ol style="list-style-type: none"> <li>1. Pengguna masuk ke halaman persembahan dari aplikasi lalu menekan tombol Log In</li> <li>2. Pengguna mengetikkan email dan password pada masing-masing kolom yang telah disetujui oleh Super Admin.</li> <li>3. Pengguna menekan tombol “Log In”</li> <li>4. Pengguna berhasil masuk ke halaman akunnya.</li> </ol>
Skenario Alternatif:	<ol style="list-style-type: none"> <li>1. Admin Komunitas mengetikkan email yang tidak disetujui Super Admin, maka system menampilkan pesan error.</li> <li>2. Pengguna mengetikkan password yang tidak disetujui Super Admin,</li> </ol>

UC20: Log In Ojek Finder Admin	
	maka system menampilkan pesan error.

**Tabel 27 UCD Mengubah Status Verifikasi Member Komunitas**

UC21: Mengubah Status Verifikasi Member Komunitas	
Ringkasan:	Use Case ini dimaksudkan agar Admin Komunitas dapat menentukan status pada member komunitas.
Aktor:	Admin Komunitas
Prioritas:	Penting (essential)
Frekuensi:	Bisa dilakukan lebih dari satu kali
Kondisi:	Admin Komunitas telah log in aplikasi "Ojek Finder Admin"
Skenario Sukses Utama:	<ol style="list-style-type: none"> <li>1. Admin Komunitas menggeser layar ke kanan (navigation drawer) lalu menekan tombol "unverified member", maka muncul daftar member komunitas yang tidak terverifikasi.</li> <li>2. Admin Komunitas menekan profil admin yang terdapat pada daftar member yang statusnya "Unverified", maka muncul detail profil member yang dipilih.</li> <li>3. Admin Komunitas menekan tombol "Verified", maka member yang dipilih statusnya telah terverifikasi sehingga mendapat nilai kepercayaan lebih dalam penggunaan aplikasi "Ojek Finder" karena telah masuk dalam suatu komunitas.</li> </ol>
Skenario Alternatif:	<ol style="list-style-type: none"> <li>1. Admin Komunitas menggeser layar ke kanan (navigation drawer) lalu</li> </ol>



UC21: Mengubah Status Verifikasi Member Komunitas	
	<p>menekan tombol “verified member”, maka muncul daftar member komunitas yang telah terverifikasi.</p> <p>2. Admin Komunitas menekan tombol “Unverified”, maka member yang dipilih statusnya tidak terverifikasi dan tidak mendapat status tergabung dalam suatu komunitas.</p>

**Tabel 28 UCD Deaktifasi Member Komunitas**

UC22: Deaktifasi Member Komunitas	
Ringkasan:	Use Case ini dimaksudkan agar Admin Komunitas dapat menghapus member komunitas.
Aktor:	Admin Komunitas
Prioritas:	Pilihan (optional)
Frekuensi:	Bisa dilakukan lebih dari satu kali
Kondisi:	Admin Komunitas telah log in aplikasi “Ojek Finder Admin”
Skenario Sukses Utama:	<ol style="list-style-type: none"> <li>1. Admin Komunitas menggeser layar ke kanan (navigation drawer) lalu menekan tombol “deaktifasi member”, maka muncul daftar member komunitas.</li> <li>2. Admin Komunitas menekan tombol profil member yang akan dideaktifasi, maka muncul window box konfirmasi penghapusan member.</li> <li>3. Admin Komunitas menekan tombol “Ya”, maka member terhapus dari daftar member.</li> </ol>

UC22: Deaktivasi Member Komunitas	
Skenario Alternatif:	<ol style="list-style-type: none"> <li>Admin Komunitas menekan tombol “Tidak”, maka member yang dipilih tidak terhapus dari daftar member.</li> </ol>

**Tabel 29 UCD Mengelola Akun Admin**

UC23: Mengelola Akun Admin	
Ringkasan:	Use Case ini dimaksudkan bahwa admin dapat merubah profil terkait nama, foto, dan nomor HP
Aktor:	Admin Komunitas
Prioritas:	Penting (essential)
Frekuensi:	Bisa lebih dari satu kali
Kondisi:	Admin telah sign up atau log in aplikasi Ojek Finder Admin
Skenario Sukses Utama:	<ol style="list-style-type: none"> <li>Admin menggeser halaman utama ke kanan, muncul tampilan Profil</li> <li>Admin menekan foto profil lalu terintegrasi ke photo library pada handphone, pengguna memilih foto dan mengukur (crop) sesuai dengan yang diinginkan, lalu klik tombol pilih.</li> <li>Admin menekan ikon edit pada nama profil, maka pengguna dapat mengedit nama profil.</li> <li>Admin menekan ikon edit pada informasi nomor HP, maka pengguna dapat mengedit nomor HP yang sesuai.</li> <li>Admin menekan tombol “Simpan” setelah selesai mengedit. Maka profil diperbarui.</li> </ol>
Skenario Alternatif:	<ol style="list-style-type: none"> <li>Admin memasukkan angka ke</li> </ol>

UC23: Mengelola Akun Admin	
	<p>dalam kolom nama, maka system akan menampilkan pesan error saat pengguna telah menekan tombol Simpan.</p> <ol style="list-style-type: none"> <li>Admin memasukkan huruf ke dalam kolom nomor HP, maka system akan menampilkan pesan error saat pengguna telah menekan tombol Simpan.</li> <li>Admin menekan tombol “Kembali” sebelum menekan tombol “Simpan”. Maka profil tidak diperbarui.</li> </ol>

**Tabel 30 UCD Mengelola Komunitas**

UC24: Mengelola Komunitas	
Ringkasan:	Use Case ini dimaksudkan bahwa admin dapat merubah profil mengenai komunitas.
Aktor:	Admin Komunitas
Prioritas:	Pilihan (Optional)
Frekuensi:	Bisa lebih dari satu kali
Kondisi:	Admin telah sign up atau log in aplikasi Ojek Finder Admin
Skenario Sukses Utama:	<ol style="list-style-type: none"> <li>Admin menggeser halaman utama ke kanan, muncul tampilan Profil</li> <li>Admin menekan ikon edit pada nama komunitas, maka admin dapat mengedit nama komunitas.</li> <li>Admin menekan tombol “Simpan” setelah selesai mengedit. Maka nama komunitas diperbarui.</li> </ol>
Skenario Alternatif:	<ol style="list-style-type: none"> <li>Admin menekan tombol “Kembali” sebelum menekan</li> </ol>



UC24: Mengelola Komunitas	
	tombol “Simpan”. Maka nama komunitas tidak diperbarui.

**Tabel 31 UCD Log Out Aplikasi “Ojek Finder Admin”**

UC25: Log Out Aplikasi “Ojek Finder Admin”	
Ringkasan:	Use Case ini dimaksudkan agar Admin Komunitas dapat keluar dari aplikasi “Ojek Finder Admin”.
Aktor:	Admin Komunitas
Prioritas:	Penting (essential)
Frekuensi:	Bisa dilakukan lebih dari satu kali
Kondisi:	Admin Komunitas telah log in aplikasi “Ojek Finder Admin”
Skenario Sukses Utama:	<ol style="list-style-type: none"> <li>Admin Komunitas menggeser layar ke kanan (navigation drawer) lalu menekan tombol “log out”, maka muncul halaman log in pada aplikasi “Ojek Finder Admin”</li> </ol>
Skenario Alternatif:	-

**Tabel 32 UCD Log In Root Admin**

UC26: Log In Root Admin	
Ringkasan:	Use Case ini dimaksudkan agar Root Admin dapat mengakses parse.com yang bertujuan untuk mengelola komunitas dan super admin.
Aktor:	Root Admin
Prioritas:	Penting (essential)
Frekuensi:	Dilakukan satu kali
Kondisi:	Root Admin telah mengakses website parse.com

UC26: Log In Root Admin	
Skenario Sukses Utama:	<ol style="list-style-type: none"> <li>1. Root Admin masuk ke halaman persembahan dari parse.com lalu menekan tombol Log In</li> <li>2. Pengguna mengetikkan email dan password yang telah didaftarkan pada saat pembuatan aplikasi.</li> <li>3. Pengguna berhasil masuk ke halaman akunnya.</li> </ol>
Skenario Alternatif:	<ol style="list-style-type: none"> <li>1. Root Admin mengetikkan email yang tidak terdaftar pada parse.com, maka system menampilkan pesan error.</li> <li>2. Pengguna mengetikkan password yang tidak terdaftar pada parse.com, maka system menampilkan pesan error.</li> </ol>

**Tabel 33 UCD Mengelola Status Komunitas**

UC27: Mengelola Status Komunitas	
Ringkasan:	Use Case ini dimaksudkan agar Root Admin dapat mengelola status dari komunitas yang terdaftar.
Aktor:	Root Admin
Prioritas:	Penting (essential)
Frekuensi:	Bisa dilakukan lebih dari satu kali
Kondisi:	Root Admin telah mengakses website parse.com
Skenario Sukses Utama:	<ol style="list-style-type: none"> <li>1. Root Admin menekan profil komunitas yang telah terdaftar pada list komunitas yang telah dibuat, maka muncul window box detail profil komunitas dan button “Active” or “Inactive”.</li> <li>2. Root Admin menekan tombol</li> </ol>

UC27: Mengelola Status Komunitas	
	“Activate”, maka komunitas yang dipilih statusnya aktif dan dapat dijadikan komunitas pada aplikasi “Super Admin Komunitas”
Skenario Alternatif:	1. Root Admin menekan tombol “Inactive”, maka komunitas yang dipilih statusnya tidak aktif dan tidak dapat dijadikan komunitas pada aplikasi “Super Admin Komunitas”

**Tabel 34 UCD Mengelola Super Admin Komunitas**

UC28: Mengelola Super Admin Komunitas	
Ringkasan:	Use Case ini dimaksudkan agar Root Admin dapat mengelola akses dari “Super Admin Komunitas”.
Aktor:	Root Admin
Prioritas:	Penting (essential)
Frekuensi:	Bisa dilakukan lebih dari satu kali
Kondisi:	Root Admin telah mengakses website parse.com
Skenario Sukses Utama:	1. Root Admin menekan tombol salah satu komunitas yang telah terdaftar pada list komunitas, pada hak akses super admin komunitas root admin mengetikkan email dan password yang dapat digunakan untuk akses aplikasi “Super Admin Komunitas”
Skenario Alternatif:	1. Root Admin tidak mengetikkan email dan password pada hak akses komunitas, maka Super Admin Komunitas tersebut tidak dapat log in pada aplikasi “Super Admin



UC28: Mengelola Super Admin Komunitas	
	Komunitas”

**Tabel 35 UCD Menerima Pengajuan Komunitas Baru**

UC29: Menerima Pengajuan Komunitas Baru	
Ringkasan:	Use Case ini dimaksudkan agar Root Admin dapat menerima pengajuan komunitas dari Pengguna Aplikasi “Ojek Finder”
Aktor:	Root Admin
Prioritas:	Penting (essential)
Frekuensi:	Bisa dilakukan lebih dari satu kali
Kondisi:	Root Admin telah mengakses website parse.com
Skenario Sukses Utama:	<ol style="list-style-type: none"> <li>1. Root Admin menekan tombol “tambah komunitas”, maka muncul halaman pendaftaran komunitas.</li> <li>2. Root Admin mengetikkan nama komunitas serta hak akses super admin komunitas, maka komunitas tersebut telah terbentuk dan super admin komunitas dapat log in melalui aplikasi “Super Admin Komunitas”</li> </ol>
Skenario Alternatif:	<ol style="list-style-type: none"> <li>2. Root Admin tidak mengetikkan nama komunitas serta hak akses super admin komunitas, maka komunitas tersebut tidak terbentuk dan super admin komunitas dapat log in melalui aplikasi “Super Admin Komunitas”</li> </ol>

Tabel 36 UCD Log Out Root Admin

UC30: Log Out Root Admin	
Ringkasan:	Use Case ini dimaksudkan agar Super Root Admin dapat keluar dari website parse.com
Aktor:	Root Admin
Prioritas:	Penting (essential)
Frekuensi:	Bisa dilakukan lebih dari satu kali
Kondisi:	Root Admin telah log in website parse.com
Skenario Sukses Utama:	1. Root Admin menekan action bar “log out”, maka muncul halaman log in pada website parse.com
Skenario Alternatif:	-

#### 4.2.8 Sequence Diagram

Sequence diagram dari ojek finder meliputi alur-alur *method* dan *object* dari setiap aplikasi Ojek Finder, Admin Komunitas Ojek Finder, dan Super Admin Komunitas Ojek Finder. Pada Android tidak ada segregasi yang nyata antara *Model*, *View*, dan *Controller*. Sehingga pada *sequence diagram* ini penulis menggunakan nama *class* dan *objek* terkait. Untuk *Class* yang menggunakan suffix *Activity* atau *Fragment*, keduanya bekerja sebagai *controller* dan *view-controller*. Gambar rancangan *sequence diagram* terdapat pada lampiran A.

#### 4.2.9 Class Diagram

*Class diagram* berisi daftar *class* yang digunakan baik itu sebagai *Model*, *View*, ataupun *Controller*. Penulis membagi *class diagram* menjadi tiga bagian sesuai aplikasi-aplikasi Ojek Finder. Yaitu aplikasi Ojek Finder, Admin Komunitas,

dan Super Admin Komunitas. Detil *Class diagram* terdapat dalam lampiran B.

#### **4.2.9.1 Class Diagram Ojek Finder**

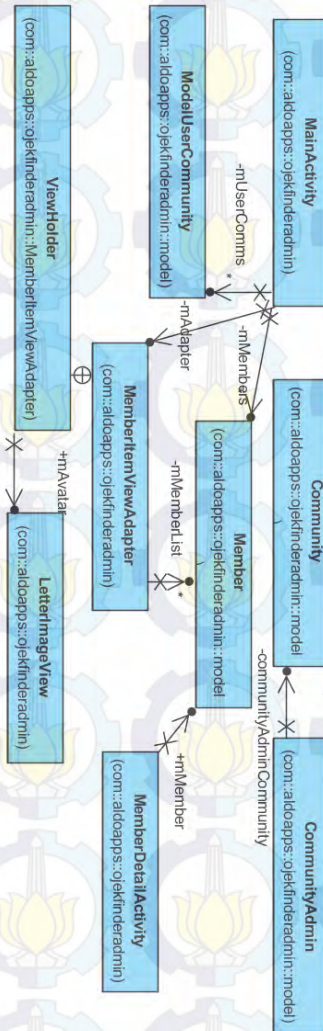
Berikut ini merupakan Class Diagram untuk aplikasi ojek finder.





#### 4.2.9.2 Class Diagram Admin Komunitas

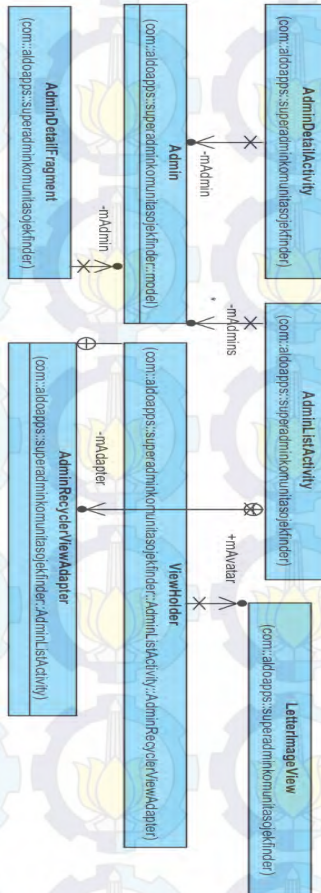
Berikut ini merupakan class diagram dari aplikasi ojek finder untuk admin komunitas.



Gambar 26 Class Diagram Admin Komunitas

#### 4.2.9.3 Class Diagram Super Admin Komunitas

Berikut ini merupakan Class Diagram untuk Super Admin Komunitas.

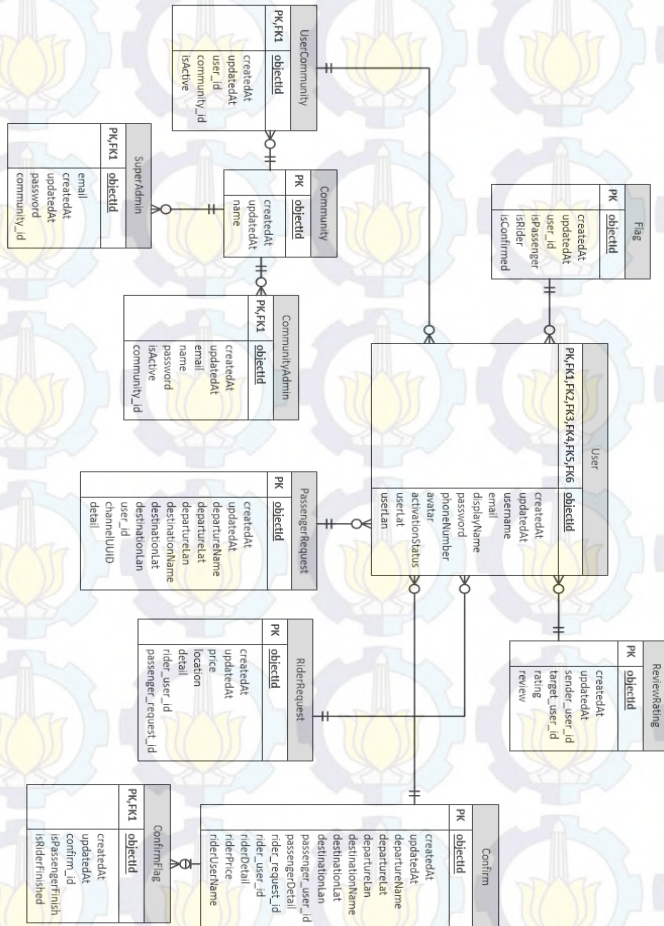


Gambar 27 Class Diagram Super Admin Komunitas



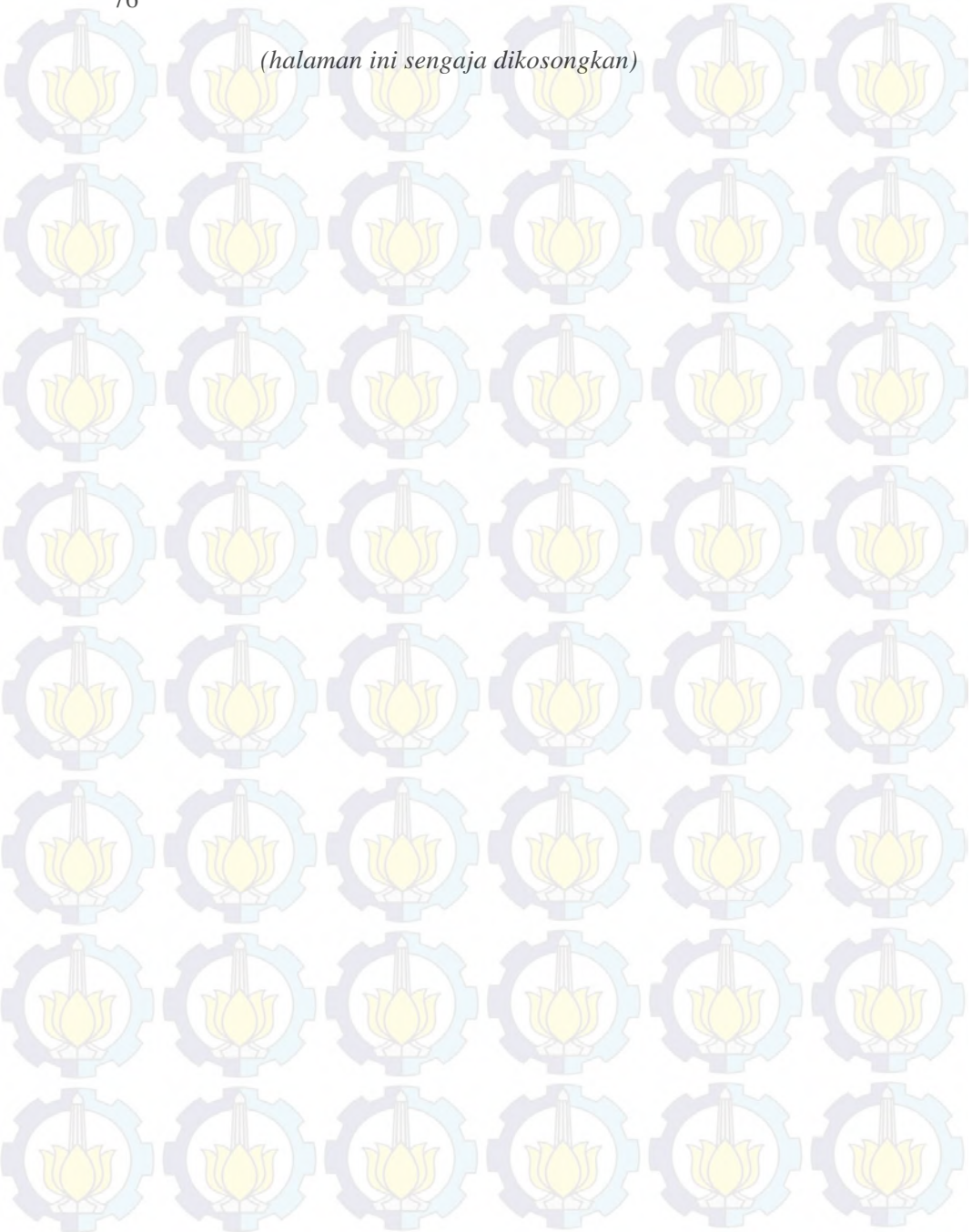
#### 4.2.10 Entity Relationship Diagram

Diagram *Entity Relationship Diagram* ini menggambarkan relasi dari setiap tabel dalam aplikasi root admin ojek finder.



Gambar 28 Skema Ojek Finder

*(halaman ini sengaja dikosongkan)*



## BAB V IMPLEMENTASI

Pada bab ini akan dijelaskan proses tahap implementasi dan ujicoba pada Aplikasi Ojek Finder, Admin Komunitas, dan Super Admin Komunitas.

### 5.1 Lingkungan Implementasi

Aplikasi Ojek Finder, Admin Komunitas, dan Super Admin Komunitas dikembangkan dengan menggunakan perangkat keras yang dapat dilihat pada Tabel 37. Sedangkan untuk perangkat lunak, aplikasi ini dikembangkan dengan menggunakan beberapa teknologi terkait dengan spesifikasi yang dapat dilihat pada Tabel 38.

**Tabel 37 Perangkat Keras Ojek Finder**

Aplikasi	Perangkat Keras	Spesifikasi
Ojek Finder, Admin Komunitas, Super Admin Komunitas	<i>Handphone</i>	Prosesor : Intel® Atom Z3560 1,8 GHz
		Memori : 2 GB
		Sistem Operasi : Android 5.0 (Marshmallow)

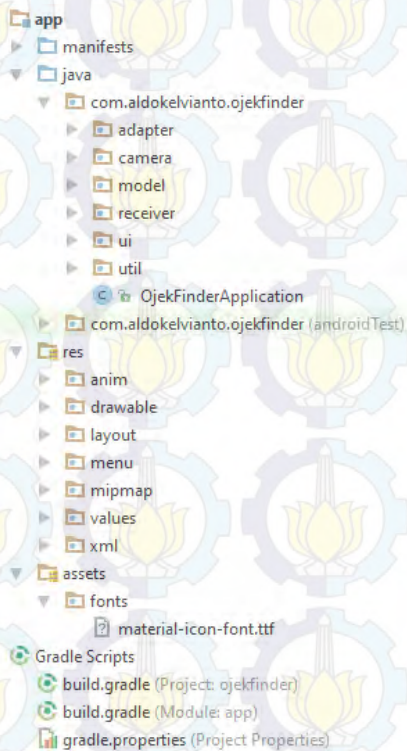
**Tabel 38 Perangkat Lunak Ojek Finder**

Aplikasi	Perangkat Lunak	Spesifikasi
Ojek Finder, Admin Komunitas, Super Admin Komunitas	Web Server	parse.com/apps/ojek-finder
	Database	NoSQL
	Bahasa Pemrograman	Java 7
	Framework Pemrograman	Android SDK
	Editor	Android Studio



## 5.2 Struktur Direktori

Struktur direktori dari aplikasi ini mengikuti standar direktori *gradle* untuk Android SDK seperti yang tertera pada Gambar 29.



Gambar 29 Sturktur Direktori Ojek Finder

## 5.3 Implementasi Ojek Finder

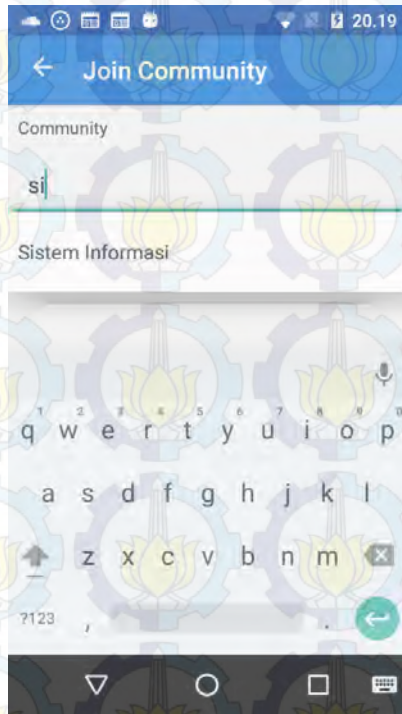
Pada bagian ini akan dijelaskan tentang bagaimana fitur-fitur utama Ojek Finder diimplementasikan dalam bentuk *source code*.

### 5.3.1 Implementasi Bergabung Dengan Komunitas

Sebelum pengguna melakukan registrasi, penulis melakukan *query* terhadap seluruh komunitas yang ada dengan nama komunitas sebagai parameternya. Potongan kode fungsi `registerToCommunity` tertera pada Gambar 30.

```
private void registerToCommunity() {  
    ParseQuery<Community> communityQuery = Community.getQuery();  
    communityQuery.whereEqualTo("name", mCommunityName.getText().toString());  
    GetCallback<Community> getCallback = (community, e) -> {  
        if (e == null) {  
            saveUserCommunity(community);  
        } else {  
            Log.d(TAG, e.getMessage());  
        }  
    };  
    communityQuery.getFirstInBackground(getCallback);  
}
```

**Gambar 30 Implementasi Bergabung Dengan Komunitas**



**Gambar 31 SS Bergabung Dengan Komunitas**

Setelah memastikan bahwa komunitas ditemukan, maka pengguna akan didaftarkan pada komunitas tersebut seperti pada *method* dalam Gambar 32.

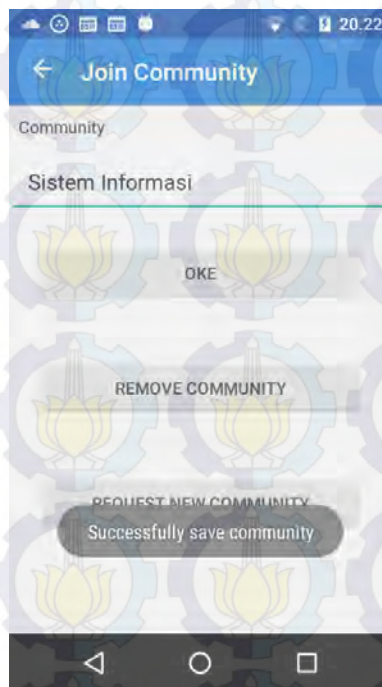


```

private void saveUserCommunity(Community community) {
    UserCommunity userCommunity = new UserCommunity();
    userCommunity.setUserObjectId(ParseUser.getCurrentUser().getObjectId());
    userCommunity.setCommunityObjectId(community.getObjectId());
    userCommunity.setIsActiveToNo();
    SaveCallback saveCallback = (e) -> {
        if (e == null) {
            Toast.makeText(getApplicationContext(),
                "Successfully save community",
                Toast.LENGTH_SHORT).show();
        } else {
            Log.d(TAG, e.getMessage());
        }
    };
    userCommunity.saveEventually(saveCallback);
}

```

**Gambar 32 Implementasi Bergabung Dengan Komunitas 2**



**Gambar 33 SS Bergabung Dengan Komunitas 2**

### 5.3.2 Implementasi melihat lokasi keberadaan semua pengguna

Untuk melihat lokasi keberadaan semua pengguna gunakan tombol dalam action bar yang akan mengaktifkan.

```
private void startLoadAllUserLocation() {  
    ParseQuery<ParseUser> query = ParseUser.getQuery();  
    query.findInBackground(new FindCallback<ParseUser>() {  
        @Override  
        public void done(List<ParseUser> list, ParseException e) {  
            if (e == null) {  
                loadAllUserLocation(list);  
            } else {  
                Log.d(TAG, e.getMessage());  
            }  
        }  
    });  
}
```

**Gambar 34 Implementasi Melihat Lokasi Keberadaan Semua Pengguna**

Setelah mendapatkan hasil *query* dari semua pengguna, penulis membuat *marker* untuk setiap lokasi pengguna tersebut seperti pada Gambar 35.

```

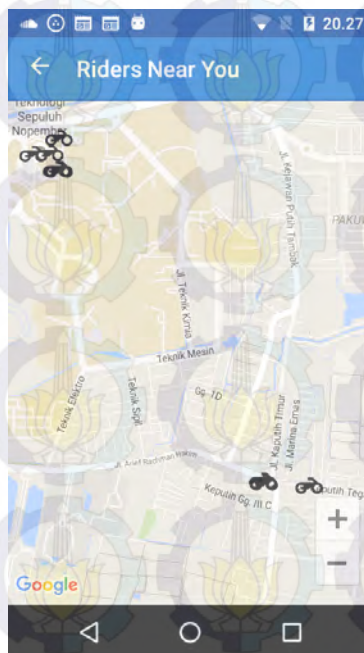
for(ParseUser parseUser : list){
    if(parseUser.getNumber("userLat") != null
        && parseUser.getNumber("userLan") != null
    ){
        double lat = parseUser.getNumber("userLat").doubleValue();
        double lan = parseUser.getNumber("userLan").doubleValue();

        MarkerOptions marker = new MarkerOptions()
            .position(new LatLng(lat, lan)).icon(icon);
        mMarkers.add(marker);

        if(parseUser.getObjectId().equals(ParseUser.getCurrentUser().getObjectId())){
            mLat = parseUser.getNumber("userLat").doubleValue();
            mLan = parseUser.getNumber("userLan").doubleValue();
        }
    }
}

```

**Gambar 35 Implementasi Melihat Lokasi Keberadaan Semua Pengguna 2**



**Gambar 36 SS Keberadaan Semua Pengguna**

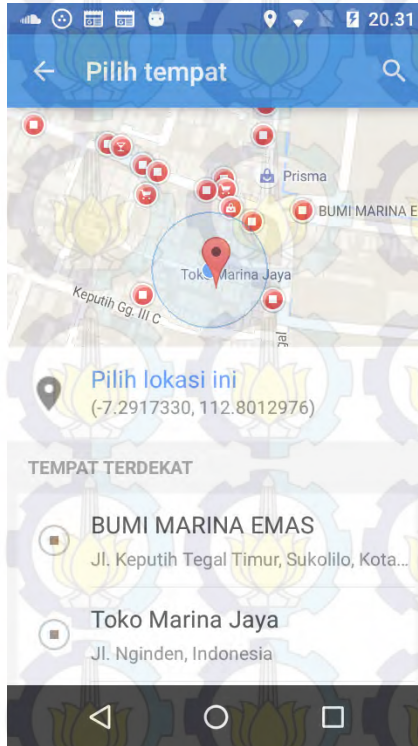


### 5.3.3 Implementasi Membuat Permintaan Tumpangan

Untuk membuat permintaan tumpangan baru, pengguna harus menginputkan lokasi keberangkatan dan tujuan. Lokasi tersebut disimpan dalam objek `PassengerRequest` beserta dengan data pengguna yang terkait, seperti pada Gambar 37.

```
PassengerRequest newPassengerReq = new PassengerRequest();  
newPassengerReq.setPassengerUser(ParseUser.getCurrentUser());  
newPassengerReq.setDepartureName(mDeparturePlace.getName().toString());  
newPassengerReq.setDepartureLat(mDeparturePlace.getLatLng().latitude);  
newPassengerReq.setDepartureLan(mDeparturePlace.getLatLng().longitude);  
newPassengerReq.setDestinationName(mDestinationPlace.getName().toString());  
newPassengerReq.setDestinationLat(mDestinationPlace.getLatLng().latitude);  
newPassengerReq.setDestinationLan(mDestinationPlace.getLatLng().longitude);  
newPassengerReq.setDetail(mExtraDetail.getText().toString());
```

**Gambar 37 Implementasi Membuat Permintaan Tumpangan**



**Gambar 38 SS Implementasi Membuat Permintaan Tumpangan**

Setelah itu penulis akan mengupdate *Flag* dari pengguna, sebagai penanda bahwa pengguna telah menjadi sub-aktor penumpang.

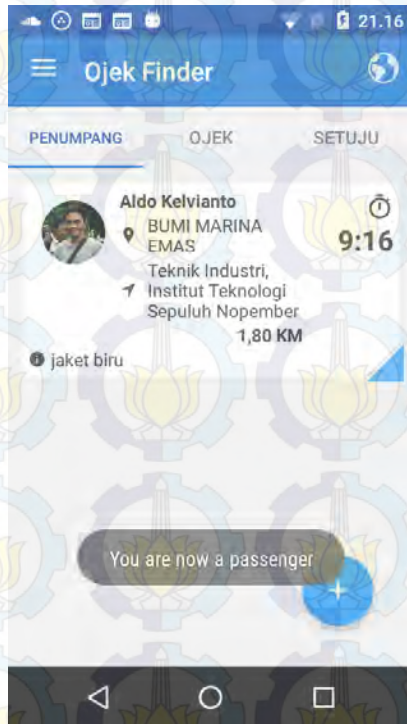
```
private void savePassengerFlagToYes(Flag flag) {  
    flag.setIsPassengerToYes();  
    SaveCallback saveCallback = new SaveCallback() {  
        @Override  
        public void done(ParseException e) {  
            if(e == null){  
                Toast.makeText(getApplicationContext(),  
                    "You are now a passenger",  
                    Toast.LENGTH_SHORT).show();  
            }else{  
                Log.d(TAG, e.getMessage());  
            }  
        }  
    };  
    flag.saveEventually(saveCallback);  
}
```

Gambar 39 Implementasi Membuat Permintaan Tumpangan 2





Gambar 40 SS Membuat Permintaan Tumpangan



**Gambar 41 SS Membuat Permintaan Tumpangan 3**

### **5.3.4 Implementasi Mengajukan Penawaran Taksi Motor**

Pada penawaran taksi motor, penulis akan mengambil nilai dari harga yang diberikan supir taksi motor beserta informasi terkait dan menyimpannya dalam objek *RiderRequest*.

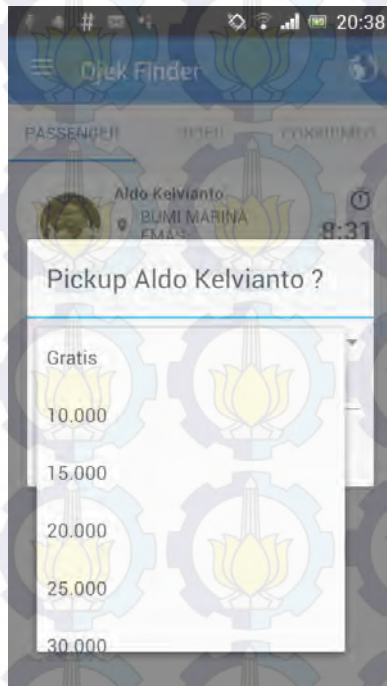
```

ParseQuery<PassengerRequest> query = PassengerRequest.getQuery();
GetCallback<PassengerRequest> getCallback = (passengerRequest, e) -> {
    riderRequest.setPassengerRequest(passengerRequest);
    riderRequest.setPrice(mSelectedPrice);
    riderRequest.setDetail(mDetail);

    SaveCallback saveCallback = (e) -> {
        if (e == null) {
            updateRiderAvailability();
        } else {
            Toast.makeText(getActivity().getApplicationContext(),
                "Gagal membuat request",
                Toast.LENGTH_SHORT).show();
        }
    };
    riderRequest.saveEventually(saveCallback);
};
query.getInBackground(mAdapter.getItem(position).getObjectId(), getCallback);

```

**Gambar 42 Implementasi Pengajuan Penawaran Taksi Motor**



**Gambar 43 SS Penawaran Taksi Motor**



Bila transaksi berhasil maka penulis akan mengubah ketersediaan pengguna menjadi seorang Supir Taksi Motor.

```
GetCallback<Flag> getCallback = (flag, e) → {
    if (e == null) {
        flag.setIsRiderToYes();
        flag.saveEventually(saveCallback);
    } else {
        Log.d(TAG, e.getMessage());
    }
};
flagQuery.getFirstInBackground(getCallback);
```

**Gambar 44 Implementasi Pengajuan Penawaran Taksi Motor**

### 5.3.5 Implementasi Menyetujui Tawarkan Taksi Motor

Ketika penumpang menyetujui tawaran taksi motor, penulis membuat objek *Confirm* yang berisikan informasi permintaan taksi motor dan permintaan tumpangan. Informasi tersebut diwujudkan dengan *objectId* yang digunakan untuk mencari kolom terkait permintaan-permintaan tersebut. lalu menghapus permintaan yang sudah dipilih.

```
Confirm confirm = new Confirm();
ModelRiderRequest rr = mAdapter.getItem(position);
PassengerRequest pr = mAdapter.getItem(position).getPassengerRequest();
mTargetRiderObjectId = rr.getRiderUser().getObjectId();

confirm.setPassengerUserObjectId(ParseUser.getCurrentUser().getObjectId());
confirm.setRiderUserObjectId(mAdapter.getItem(position).getRiderUser().getObjectId());
```

**Gambar 45 Implementasi Persetujuan Taksi Motor**



**Gambar 46 SS Menyetujui Penawaran Taksi Motor**

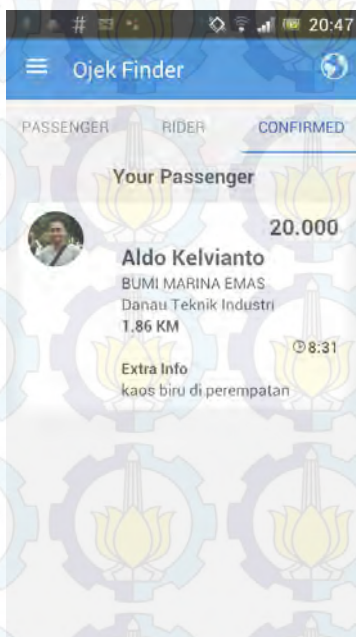
Penulis juga mengubah *ConfirmFlag* pengguna untuk menandakan bahwa saat ini pengguna sudah memiliki transaksi.

```
SaveCallback saveCallback = (e) -> {
    if (e == null) {
        updateConfirmAvailability();
        startDeleteRiderRequestsAndUpdateTheirFlag();
        startDeleteYourPassengerRequest();

        Toaster.fromFragmentSendRequestSuccess(RiderListFragment.this);
    } else {
        Toaster.fromFragmentSendRequestFail(RiderListFragment.this);
    }
};

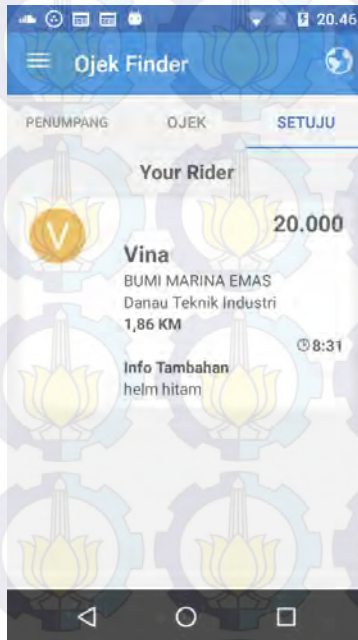
confirm.saveEventually(saveCallback);
```

**Gambar 47 Implementasi Persetujuan Taksi Motor 2**



Gambar 48 SS Menyetujui Tawaran Taksi Motor 2





**Gambar 49 SS Menyetujui Tawaran Taksi Motor 3**

### **5.3.6 Implementasi Mengakhiri Persetujuan Taksi Motor**

Ketika kedua pengguna sama-sama mengakhiri persetujuan taksi motor, maka objek *Confirm* akan dihapus dan penulis akan mengubah *ConfirmFlag* dari pengguna menjadi false. Namun sebelum penulis akan menampilkan *dialog Rating* dan *Review* terhadap pengguna.

```

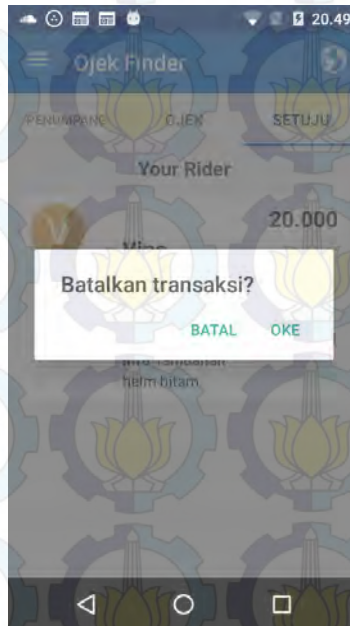
if (mFlag.isCurrentlyARider() && mFlag.isCurrentlyConfirmed()) {
    if(!confirmFlag.isPassengerFinish() && confirmFlag.isRiderFinish()){
        Toaster.fromFragment(this, "Waiting for other user to finish");
    }

    if(confirmFlag.isPassengerFinish() && !confirmFlag.isRiderFinish()){
        setUserConfirmFlagToFinish(confirmFlag);
    }
}

if(confirmFlag.isPassengerFinish() && confirmFlag.isRiderFinish()){
    deleteConfirmFlagAndConfirm();
}

```

**Gambar 50 Implementasi Mengakhiri Persetujuan Taksi Motor**



**Gambar 51 SS Implementasi Pembatalan Taksi Motor**

```

GetCallback<ConfirmFlag> getCallback = new GetCallback<ConfirmFlag>() {
    @Override
    public void done(ConfirmFlag confirmFlag, ParseException e) {
        if (e == null) {
            confirmFlag.deleteInBackground(deleteCallback);
        } else {
            Log.d(TAG, e.getMessage());
        }
    }
};
query.getFirstInBackground(getCallback);

```

**Gambar 52 Implementasi Mengakhiri Persetujuan Taksi Motor**  
2

### 5.3.7 Implementasi Memberikan Rating dan Review

Penulis membedakan *rating* dan *review* dari status pengguna, apakah dia penumpang atau supir taksi motor. Setelah itu barulah nilai dari *rating* dan *review* dikirim.

```

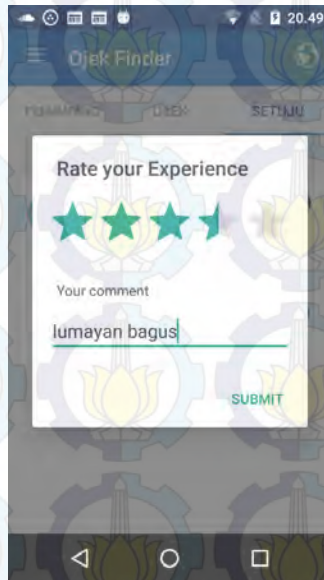
ReviewRating rr = new ReviewRating();
rr.setReview(commentText);
rr.setRating(ratingValue);

if(mFlag.isCurrentlyAPassenger() && mFlag.isCurrentlyConfirmed()){
    rr.setTargetObjectId(mConfirm.getRiderUserObjectId());
    rr.setSenderId(ParseUser.getCurrentUser().getObjectId());
}

```

**Gambar 53 Implementasi Memberikan Rating dan Review**





**Gambar 54 SS Memberikan Rating dan Review**

```
SaveCallback saveCallback = (e) -> {
    if (e == null) {
        Toaster.fromFragmentSendReviewSuccess(ConfirmedListFragment.this);
    } else {
        Log.d(TAG, e.getMessage());
    }
};

rr.saveInBackground(saveCallback);
```

**Gambar 55 Implementasi Memberikan Rating dan Review 2**

### 5.3.8 Implementasi Mengaktivasi dan Deaktivasi Anggota

Pada Aplikasi Ojek Finder untuk Admin Komunitas, penulis mengubah status aktivasi pengguna pada objek *userCommunity*.

```

private void updateUserCommunity(UserCommunity userCommunity) {
    if (mMember.getStatus().equalsIgnoreCase("yes")) {
        userCommunity.setIsActiveToNo();
    } else {
        userCommunity.setIsActiveToYes();
    }
    SaveCallback saveCallback = (e) -> {
        if (e == null) {
            Toast.makeText(getApplicationContext(),
                "Successfully updated user status!",
                Toast.LENGTH_SHORT).show();
        } else {
            Log.d(TAG, e.getMessage());
        }
    };
    userCommunity.saveEventually(saveCallback);
}

```

**Gambar 56 Implementasi Mengaktivasi dan Deaktivasi Anggota**



**Gambar 57 SS Aktivasi dan Deaktivasi Anggota**

### 5.3.9 Implementasi Pembuatan Admin Baru

Untuk pembuatan admin yang baru pada Aplikasi Super Admin Komunitas, penulis akan membuat objek Admin dan menyimpan atribut terkait seperti email dan password.

```
ParseObject currentAdmin = list.get(0);
ParseObject suCommunity = currentAdmin.getParseObject("superAdminCommunity");
ParseObject newAdmin = new ParseObject("CommunityAdmin");
newAdmin.put("email", email);
newAdmin.put("password", password);
newAdmin.put("name", name);
newAdmin.put("isActive", "yes");
newAdmin.put("communityAdminCommunity", suCommunity);
newAdmin.saveEventually(saveCallback);
finish();
```

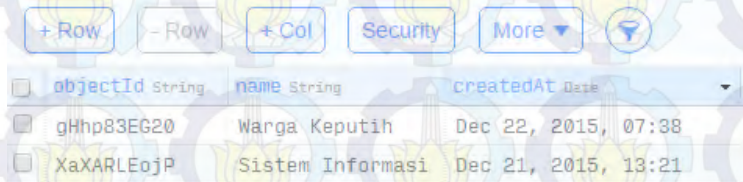
**Gambar 58 Implementasi Pembuatan Admin Baru**

**Gambar 59 SS Pembuatan Admin Baru**



### 5.3.10 Implementasi Pembuatan Komunitas dan Super Admin Komunitas

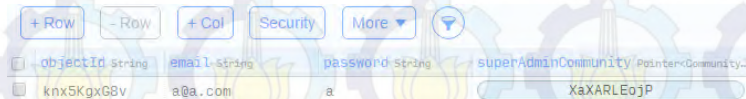
Untuk membuat Komunitas dan Super Admin Komunitas yang baru, *Root* admin akan menambahkan baris baru pada tabel Community dan SuperAdmin.



<input type="checkbox"/>	objectId String	name String	createdAt Date
<input type="checkbox"/>	gHhp83EG20	Warga Keputih	Dec 22, 2015, 07:38
<input type="checkbox"/>	XaXARLEojP	Sistem Informasi	Dec 21, 2015, 13:21

**Gambar 60 Implementasi Pembuatan Komunitas dan Super Admin Komunitas**

Super Admin akan mencantumkan *reference* berupa *pointer* terhadap objectId Community.



<input type="checkbox"/>	objectId String	email String	password String	superAdminCommunity/Pointer~Community...
<input type="checkbox"/>	knx5KgXGBv	a@a.com	a	XaXARLEojP

**Gambar 61 Implementasi Pembuatan Komunitas dan Super Admin Komunitas 2**

## 5.4 Pengujian

Pada bagian ini akan dilakukan pengujian menggunakan Uji Coba Fungsional dan *User Acceptance Test*.

### 5.4.1 Uji Coba Fungsional

**Tabel 39 Format Test Case**

No	Aksi Test	Data Test	Hasil yang Diharapkan	Hasil Aktual	Sukses / Gagal
1	Akis yang dilakukan Pengguna	Data yang diisikan pengguna	Hasil yang terjadi dalam aplikasi	Hasil yang dilihat oleh pengguna	Sukses

Detil lengkap dari setiap *Test Case* terdapat pada Lampiran C.

**Tabel 40 Daftar Test Case**

Kode Test Case	Nama Test Case
TC-01	Sign Up
TC-02	Login
TC-03	Logout
TC-04	Mengganti Nama dan No. HP
TC-05	Mengganti <i>profile picture</i>
TC-06	Bergabung ke Komunitas
TC-07	Keluar Komunitas
TC-08	Melihat Keberadaan Semua Pengguna
TC-09	Memberikan Review dan Rating
TC-10	Membatalkan Transaksi
TC-11	Membuat Permintaan Tumpangan Baru
TC-12	Menyetujui Penawaran Taksi Motor
TC-13	Membuat Tawaran Taksi Motor
TC-14	Mengaktivasi dan Deaktivasi Pengguna
TC-15	Membuat Admin Baru
TC-16	Mengaktivasi dan Deaktivasi Admin Komunitas

### 5.4.2 User Acceptance Test

*User Acceptance Test* (UAT) digunakan untuk mengetahui apakah hasil dari implementasi *functional requirement* telah dirasa cukup oleh calon pengguna aplikasi sebelum diluncurkan untuk umum. UAT dibuat berdasarkan format

dokumen dari *Swiburne University of Technology* dan menggunakan skala likert untuk jawaban dari responden. Responden akan diberikan instruksi untuk menggunakan aplikasi dan selanjutnya dilepas untuk menggunakan aplikasi tanpa instruksi sembari menuliskan *feedback* UAT. Berikut merupakan format dari UAT yang diberikan kepada responden.

UAT Ojek Finder			
No	Task Item	Tidak Puas	Puas
<b>Pengguna</b>			
1	Sign Up		
	Komentar :		
2	Login dan Logout		
	Komentar :		
3	Mengganti Nama dan No. HP		
	Komentar :		
4	Mengganti Profil Picture		
	Komentar :		
5	Gabung dan keluar dari komunitas		
	Komentar :		
6	Melihat keberadaan semua pengguna		
	Komentar :		
7	Memberikan Rating dan Review		
	Komentar :		
8	Membatalkan transaksi		
	Komentar :		
<b>Penumpang</b>			
1	Mneginput lokasi keberangkatan dan tujuan		
	Komentar :		



UAT Ojek Finder			
No	Task Item	Tidak Puas	Puas
2	Memilih tawaran taksi motor		
	Komentar :		
<b>Supir Taksi Motor</b>			
1	Mengajukan penawaran taksi motor		
	Komentar :		
<b>Admin Komunitas</b>			
1	Mengaktivasi dan deaktivasi pengguna		
	Komentar :		
<b>Super Admin Komunitas</b>			
1	Membuat admin baru		
	Komentar :		
2	Mengaktivasi dan deaktivasi Admin Komunitas		
	Komentar :		

Tabel 41 Format UAT

## BAB VI HASIL DAN PEMBAHASAN

Pada bab ini akan dijelaskan apa saja hasil yang didapatkan dari pengujian beserta analisa dari pengujian tersebut.

### 6.1 Hasil Pengujian

Pada bagian ini akan menampilkan hasil dari Uji Coba Fungsional dan *User Acceptance Test*.

#### 6.1.1 Uji Coba Fungsional

Berikut ini merupakan hasil dari pengujian Uji Coba Fungsional sesuai yang telah disepakati pada Bab 4.

**Tabel 42 Hasil Uji Coba Fungsional**

Kode Test Case	Nama Test Case	Status
TC-01	Sign Up	Terpenuhi
TC-02	Login	Terpenuhi
TC-03	Logout	Terpenuhi
TC-04	Mengganti Nama dan No. HP	Terpenuhi
TC-05	Mengganti <i>profile picture</i>	Terpenuhi
TC-06	Bergabung ke Komunitas	Terpenuhi
TC-07	Keluar Komunitas	Terpenuhi
TC-08	Melihat Keberadaan Semua Pengguna	Terpenuhi
TC-09	Memberikan Review dan Rating	Terpenuhi
TC-10	Membatalkan Transaksi	Terpenuhi
TC-11	Membuat Permintaan Tumpangan Baru	Terpenuhi
TC-12	Menyetujui Penawaran Taksi Motor	Sebagian
TC-13	Membuat Tawaran Taksi Motor	Terpenuhi
TC-14	Mengaktivasi dan Deaktivasi Pengguna	Terpenuhi
TC-15	Membuat Admin Baru	Terpenuhi
TC-16	Mengaktivasi dan Deaktivasi Admin Komunitas	Terpenuhi

### 6.1.2 Hasil Pengujian User Acceptance Test

Hasil dari pengujian *User Acceptance Test* terhadap sembilan orang responden. Selain mengisikikan kepuasan responden juga memberikan komentar dan saran terhadap aplikasi.

**Tabel 43 Hasil UAT**

UAT Ojek Finder			
No	Task Item	Tidak Puas	Puas
<b>Pengguna</b>			
1	Sign Up		9
2	Login dan Logout	1	8
3	Mengganti Nama dan No. HP		9
4	Mengganti Profil Picture		9
5	Gabung dan keluar dari komunitas	2	7
6	Melihat keberadaan semua pengguna	1	8
7	Memberikan Rating dan Review		9
8	Membatalkan transaksi	3	7
<b>Penumpang</b>			
1	Menginput lokasi keberangkatan dan tujuan	1	8
2	Memilih tawaran taksi motor		9
<b>Supir Taksi Motor</b>			
1	Mengajukan penawaran taksi motor		9
<b>Admin Komunitas</b>			
1	Mengaktivasi dan deaktivasi pengguna	2	7
<b>Super Admin Komunitas</b>			
1	Membuat admin baru	1	8
2	Mengaktivasi dan deaktivasi Admin Komunitas	2	7



UAT Ojek FINDER			
No	Task Item	Tidak Puas	Puas
Total		13	114

Tabel 44 Komentar UAT

No	Komentar dan Saran
<b>Pengguna</b>	
1	Pada saat membatalkan transaksi kurang auto-refresh
2	Gambar motornya cari yang agak bagus
3	Jangan menggunakan istilah cancel untuk transaksi yang berhasil atau selesai. Ambigu dengan cancel (batal)
4	Cukup jelas, tapi belum ada opsi lupa password
5	Ada namun sedikit sulit untuk menemukan menu
6	Sudah bagus, ada opsi mengambil gambar dari galeri atau dari kamera. Mungkin jika bisa diintegrasikan dengan social media bisa otomatis mengambil gambar dari media sosial
7	Fungsi komunitas masih kurang jelas selain hanya untuk verifikasi driver (adakah layanan / fitur tambahan untuk penumpang jika mendaftar di komunitas?)
8	Err, ini pengguna atau driver yang dilihat keberadaannya? Kalau driver sudah cukup bagus
9	Sulit membatalkan transaksi jika hanya satu pihak yang ingin batal
<b>Penumpang</b>	
1	Menu untuk menambahkan 'order' baru tersembunyi dan hanya berupa tanda plus sehingga cukup membingungkan. Pointer penunjuk tipenya fixed, sedikit membingungkan
2	Tampilan diurutkan berdasarkan waktu 'pengambilan', apa bisa diurutkan berdasarkan harga? Atau mungkin lokasi driver?

No	Komentar dan Saran
3	Kurang auto-refresh
4	Kurang notifikasi
Admin Komunitas	
1	Masih bisa ditingkatkan UI-nya
2	Info user perlu lebih detail biar bisa disetujui dan dipercayai oleh admin komunitas
3	Aktifasi dan deaktivasi membingungkan, karena ketika status berubah tidak tampak
Super Admin Komunitas	
1	Menu untuk membuat admin baru tersembunyi
2	Aktifasi dan deaktivasi membingungkan, karena status berubah tidak tampak
3	Ada notif ke member kalau dia diaktivasi atau dideaktivasi
4	Masih bingung seputar komunitas

## 6.2 Analisa Pengujian

Pada bagian ini akan dijelaskan analisa dari hasil Uji Coba Fungsional dan *User Acceptance Test*.

### 6.2.1 Analisa Uji Coba Fungsional

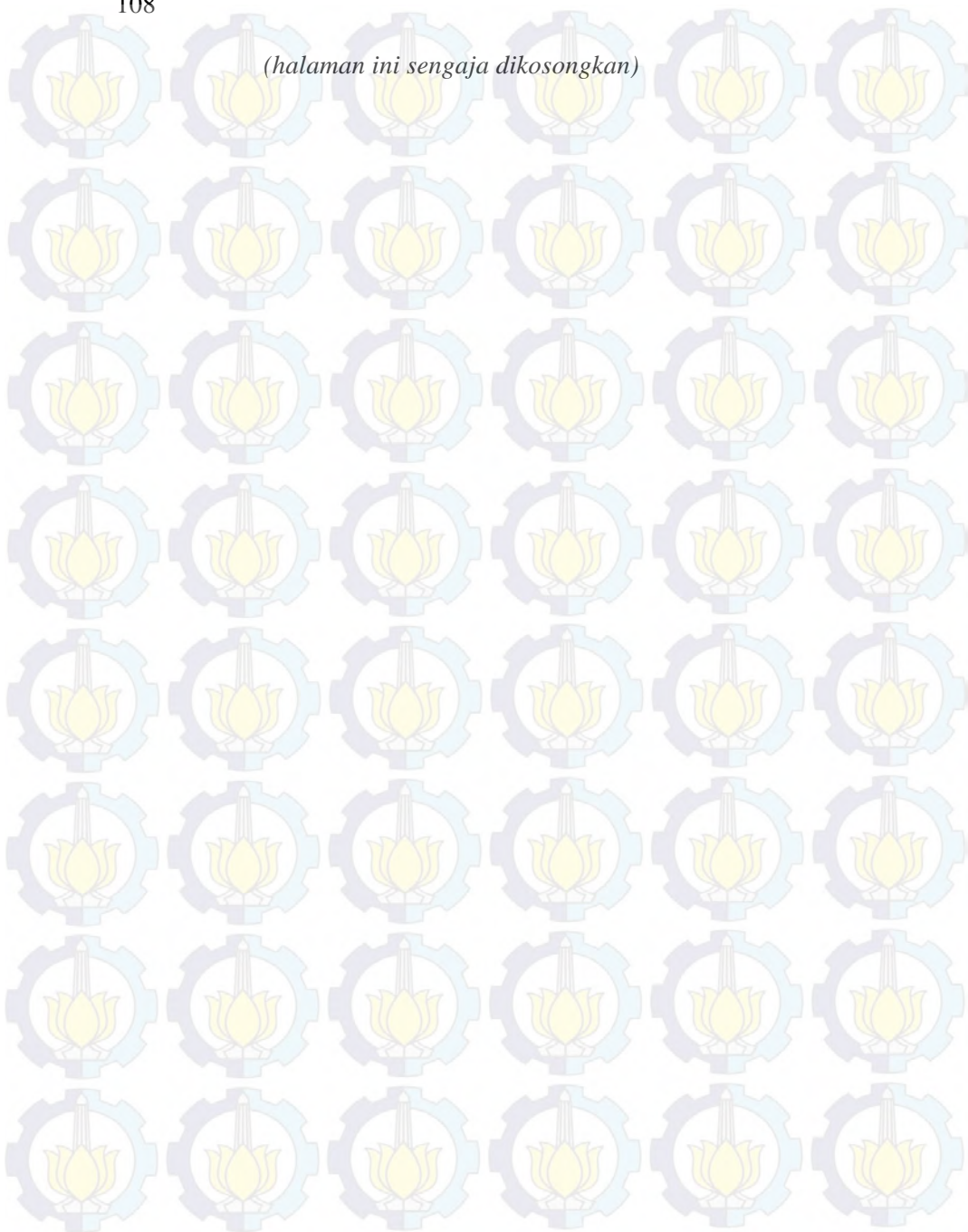
Dalam pengujian ini semua input yang dimasukan merupakan kondisi optimal, tanpa *stress condition*, dimana pengguna memasukan input yang tidak seharusnya. Dan dari hasil pengujian diatas dapat dilihat bahwa hampir semua fitur *test case* dari aplikasi terpenuhi. Pada test-case TC12 dari beberapa kejadian pengujian, dari lima hingga sepuluh kali percobaan, ada satu kali kondisi yang menyebabkan *force close*. Namun kondisi tersebut belum bisa di *reproduce* sehingga sulit untuk diperbaiki. Untuk test-case yang lain semuanya berjalan normal.

### 6.2.2 Analisis Pengujian User Acceptance Test

Berdasarkan hasil pengujian *User Acceptance Test* terhadap sembilan responden dapat dilihat bahwa terdapat 114 jawaban puas dan 13 jawaban tidak puas. Kebanyakan responden yang menjawab sangat puas dan puas telah mencoba fitur secara maksimal bahkan bereksperimen dengan fitur tersebut, seperti mencoba membuat dua kali permintaan tumpangan. Sedangkan responden yang menjawab tidak puas dan sangat tidak puas merasa kurang terbiasa dengan lingkungan *UI Android* seperti *Floating Action Button* dan aspek desain lainnya. Bila diperhitungkan rata-rata bisa disimpulkan bahwa responden merasa 90% puas terhadap fitur aplikasi Ojek Finder, Admin Komunitas Ojek Finder, dan Super Admin Komunitas Ojek Finder.



*(halaman ini sengaja dikosongkan)*



## BAB VII

### KESIMPULAN DAN SARAN

Pada bab ini ada dijelaskan kesimpulan dan saran dari seluruh pengerjaan tugas akhir. Kesimpulan dan saran diharapkan berguna untuk proses pengembangan aplikasi selanjutnya.

#### 7.1 Kesimpulan

Berdasarkan pengerjaan tugas akhir ini, didapatkan beberapa kesimpulan sebagai berikut:

1. Konsep pembuatan aplikasi untuk mencari taksi motor berbasis komunitas berhasil dibuat. Dengan membuat tiga aplikasi yang berbeda untuk pengguna, admin dan super admin. Pengguna bisa menggunakan aplikasi Ojek Finder untuk menjadi penumpang dan supir taksi motor. Admin komunitas menggunakan aplikasi Ojek Finder Admin untuk memverifikasi pengguna yang ingin bergabung dengan komunitas, dan mengubah profil komunitas. Dan super admin komunitas menggunakan aplikasi Super Admin Komunitas untuk menunjuk siapa saja Admin yang berhak untuk mengelola komunitas.
2. Respon dari responden terhadap aplikasi ojek finder cukup baik dimana 90% dari responden merasa puas terhadap layanan ojek finder. Sementara dari segi penyelesaian fitur, 94% sudah berfungsi dengan baik. Fitur yang masih memiliki masalah adalah ketika penumpang menyetujui supir taksi motor, ada kalanya aplikasi memunculkan *null pointer exception*. Namun hal itu hanya terjadi satu kali dari lima hingga – sepuluh kali percobaan.
3. Aplikasi dapat berjalan dengan baik pada Android 4.0 yang merupakan *device* lama, sehingga memungkinkan masyarakat untuk menggunakan aplikasi ojek finder tanpa membeli *device* yang terbaru.

## 7.2 Saran

Pada bagian ini akan dipaparkan beberapa kekurangan pada aplikasi ojek finder yang perlu diperbaiki dan dikembangkan lagi. Berikut ini masukan untuk penelitian selanjutnya.

1. Sesuai dengan komentar dan saran dari responden *User Acceptance Test* penulis akan menambahkan fitur-fitur yang cukup membantu yaitu, pemberian notifikasi terhadap penumpang yang menunggu tawaran taksi motor, supir taksi motor yang diterima dan ditolak tawarannya oleh penumpang, dan penambahan *auto-refresh* terhadap daftar tawaran tumpangan dan taksi motor.
2. Dari segi *wording*, aplikasi ini juga perlu standardisasi dan perbaikan kata-kata untuk setiap transaksi yang dilakukan oleh pengguna. Karena beberapa kata-kata masih belum baku dan mudah dimengerti oleh pengguna.
3. Performa aplikasi dan stabilitas aplikasi terhadap *stress testing* juga perlu dipertimbangkan untuk penelitian selanjutnya. Aplikasi harus bisa berjalan dengan kondisi yang tidak ideal seperti masukan pengguna yang tidak sesuai atau kondisi *device* tanpa internet.
4. Beberapa responden juga memberikan saran untuk mengadakan fitur multi-komunitas dimana supir taksi motor dapat mendaftar tidak hanya dalam satu komunitas namun lebih dari satu komunitas.



## DAFTAR PUSTAKA

1. Ramadhan A. Antara News. [Online].; 2015 [cited 2015 1 9. Available from: <http://www.antaraneews.com/berita/473169/jumlah-motor-dan-mobil-di-jakarta-tumbuh-12-persen-tiap-tahun>.
2. Rakyat DPUdP. Dinas Pekerjaan Umum dan Perumahan Rakyat. [Online].; 2015. Available from: [http://pu.go.id/main/view\\_pdf/7843](http://pu.go.id/main/view_pdf/7843).
3. Kompas HTHdKH. Historia. [Online]. Available from: <http://historia.id/kota/mengorek-sejarah-ojek>.
4. Hutabarat LC. Metro TV. [Online].; 2015. Available from: <http://news.metrotvnews.com/read/2015/06/30/141847/pendaftaran-membludak-gojek-batasi-penerimaan-driver>.
5. Google. Google Play. [Online].; 2015. Available from: <https://play.google.com/store/apps/details?id=com.gojek.app>,  
<https://play.google.com/store/apps/details?id=com.customer.blujek>,  
<https://play.google.com/store/apps/details?id=ladyjek.twiscode.com.driver>,  
<https://play.google.com/store/apps/details?id=com.gr>.
6. Kuncoro M. Kabel. [Online].; 2015. Available from: <http://www.kabel.co.id/urair-kemacetan-jakarta-ahok-wacanakan-inetgrasikan-busway-dan-gojek/1230/>.
7. Sus. OkeZone. [Online].; 2015. Available from: <http://news.okezone.com/read/2015/06/18/338/1167384/go-jek-bisa-atasi-kemacetan-di-jakarta>.
8. GO-JEK. [Online].; 2014. Available from:

- <http://www.go-jek.com/>.
9. Grab Taxi. [Online].; 2012. Available from: <http://grabtaxi.com/jakarta-indonesia/tentang-kami/>.
  10. James P. Globalism, Nationalism, Tribalism: Bringing Theory Back In —Volume 2 of Towards a Theory of Abstract Community London: Sage; 2006.
  11. Wenger E, Snyder , McDermott R. Cultivating Communities of Practice: Harvard Business Press; 2002.
  12. Belk R. You are what you can access: Sharing and collaborative consumption online. Business Research. 2014;; p. 1595-1600.
  13. American Dialect. [Online].; 2010 [cited 2010 January. Available from: <http://www.americandialect.org/app-voted-2010-word-of-the-year-by-the-american-dialect-society-updated>.
  14. The Guardian. [Online].; 2013 [cited 2013 September. Available from: <http://www.theguardian.com/technology/appsblog/2013/sep/19/gartner-mobile-apps-revenues-report>.
  15. Survey NHT. U.S. Department of Transportation, Federal Highway Administration. [Online].; 2009. Available from: <http://nhts.ornl.gov/tables09/ae/work/Job24716.html>.
  16. Ljiljana Zekanović K, Grzunov J. Evaluation of shared economy adoption: Case of Airbnb.com. 2014.
  17. Wired. [Online].; 2014 [cited 2014 11. Available from: <http://www.wired.com/insights/2014/11/hotels-sharing-economy/>.
  18. Gil P. About. [Online].; 2014 [cited 2014 December.

Available from:  
[http://netforbeginners.about.com/od/s/f/what\\_is\\_SaaS\\_software\\_as\\_a\\_service.htm](http://netforbeginners.about.com/od/s/f/what_is_SaaS_software_as_a_service.htm).

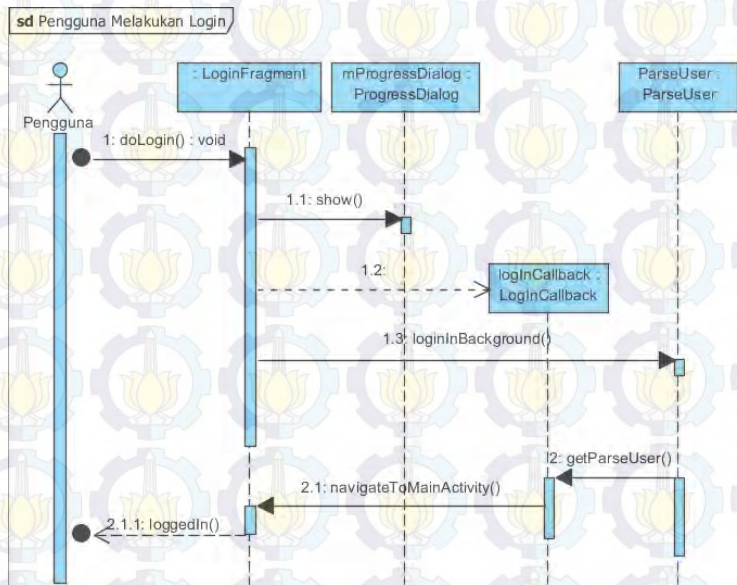
19. Matthias Hirth THPTG. Anatomy of a Crowdsourcing Platform - Using the Example of Microworkers.com. IEEE International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing. 2011 June;(5).

20. Zipcar. [Online]. Available from:  
<http://www.zipcar.com/>.



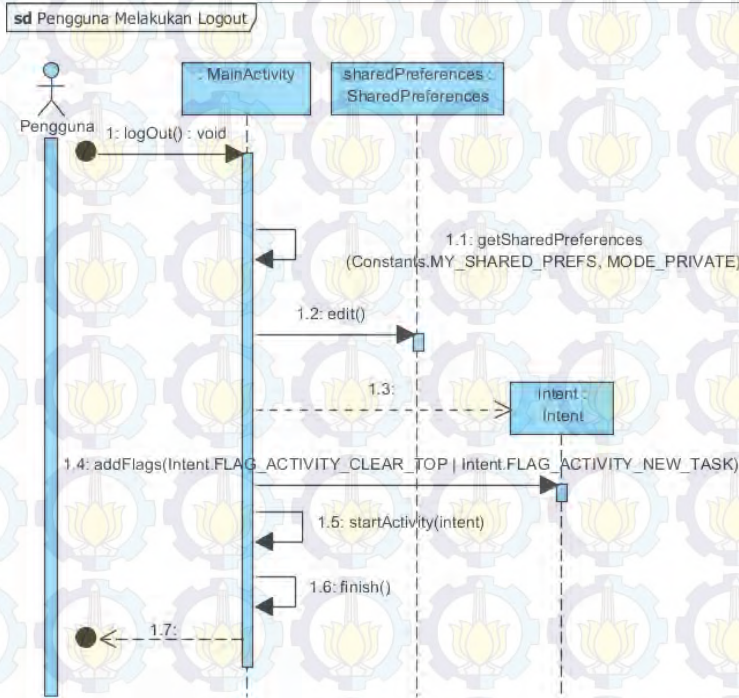
## LAMPIRAN A SEQUENCE DIAGRAM

### A.1 Sequence Diagram Untuk Pengguna



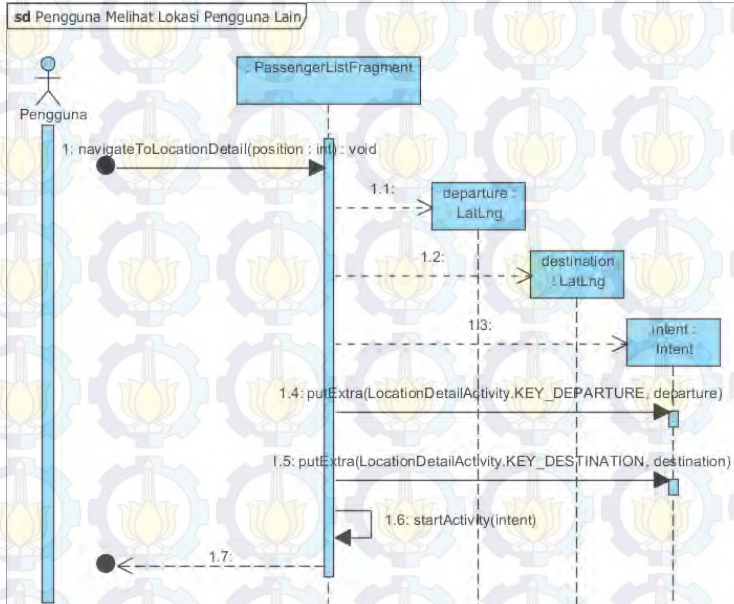
Gambar 62 SD Pengguna Melakukan Login

A-2



**Gambar 63 SD Pengguna Melakukan Logout**



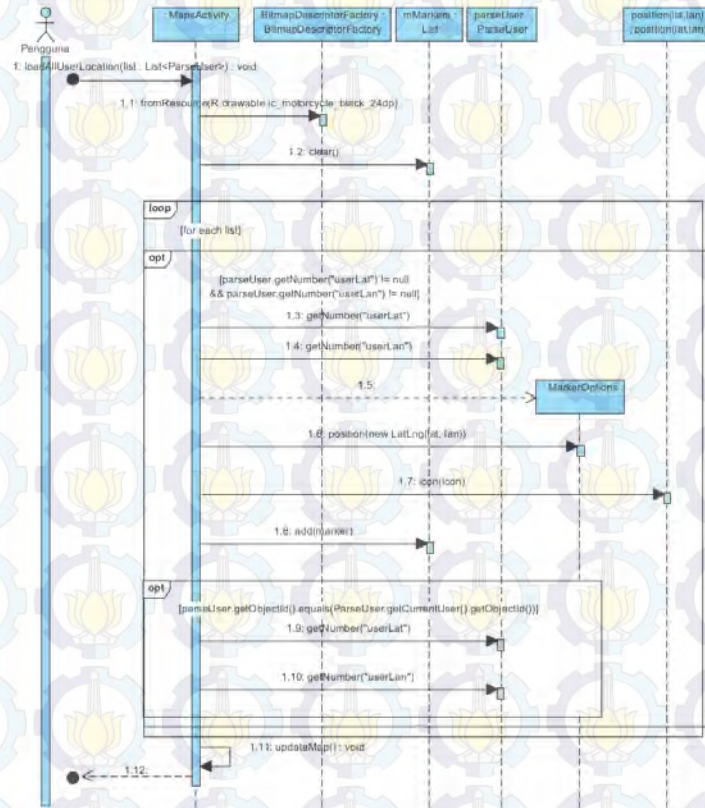


**Gambar 64 SD Pengguna Melihat Lokasi Pengguna Lain**

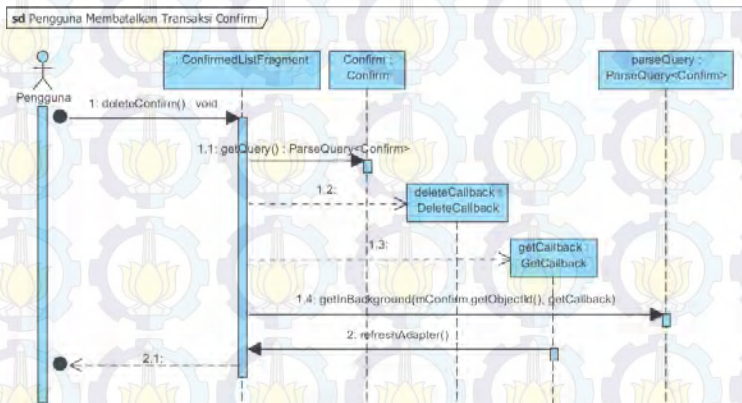


A-4

sd Pengguna Melihat Lokasi Pengguna Di Sekitarnya

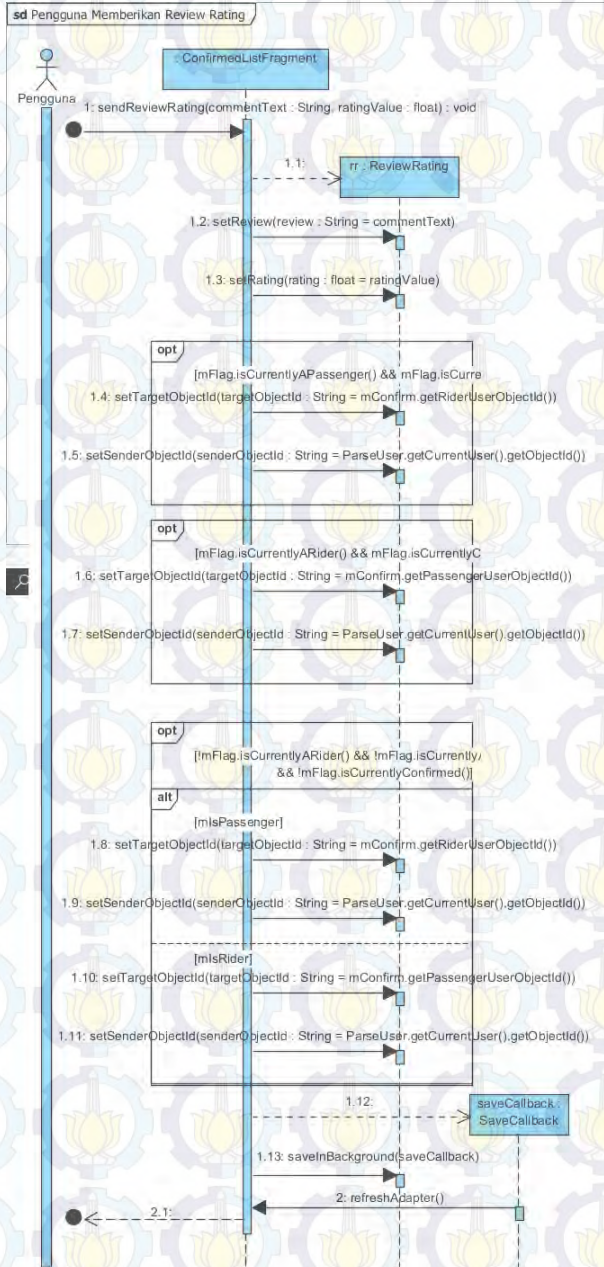


**Gambar 65 SD Pengguna Melihat Lokasi Pengguna Di Sekitarnya**



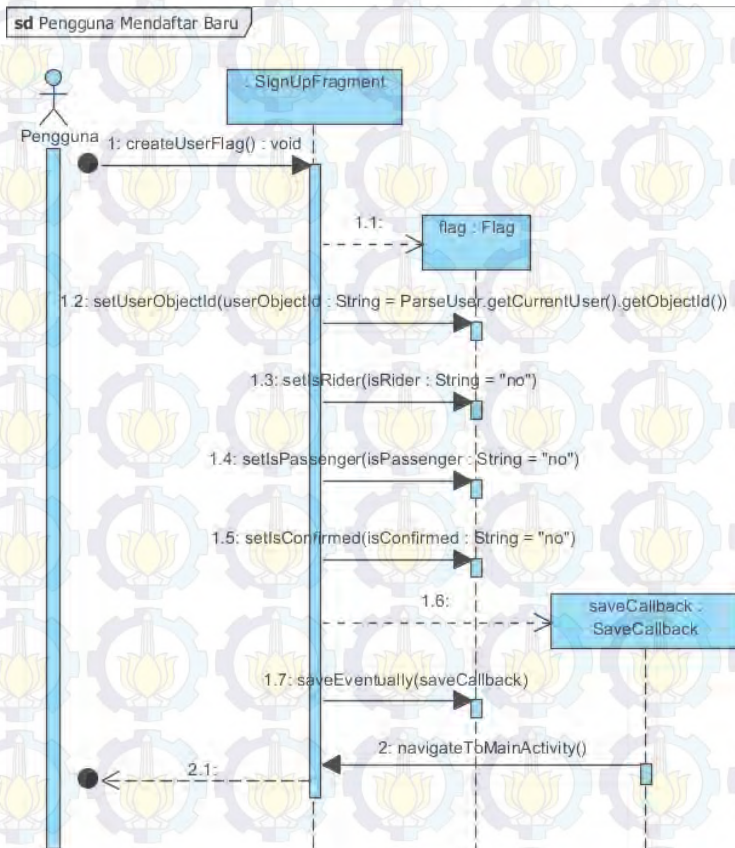
**Gambar 66 SD Pengguna Membatalkan Transaksi Confirm**





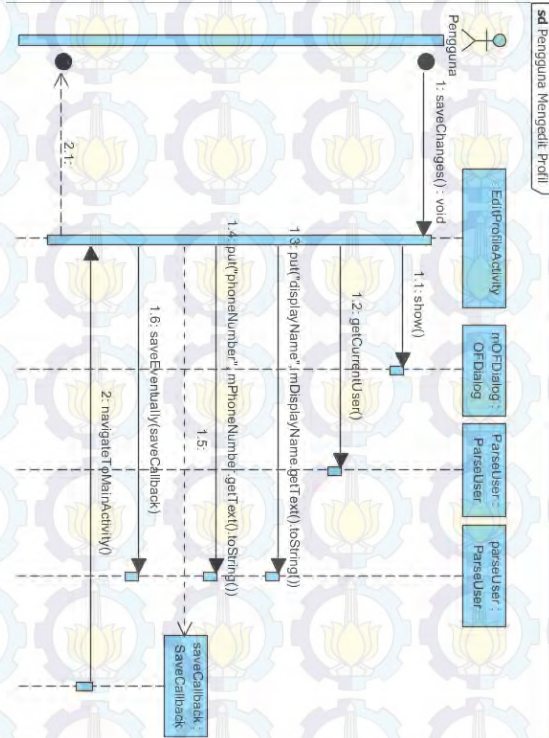


Gambar 67 SD Pengguna Memberikan Review Rating



Gambar 68 SD Pengguna Mendaftar Baru

A-8



Gambar 69 SD Pengguna Mengedit Profil



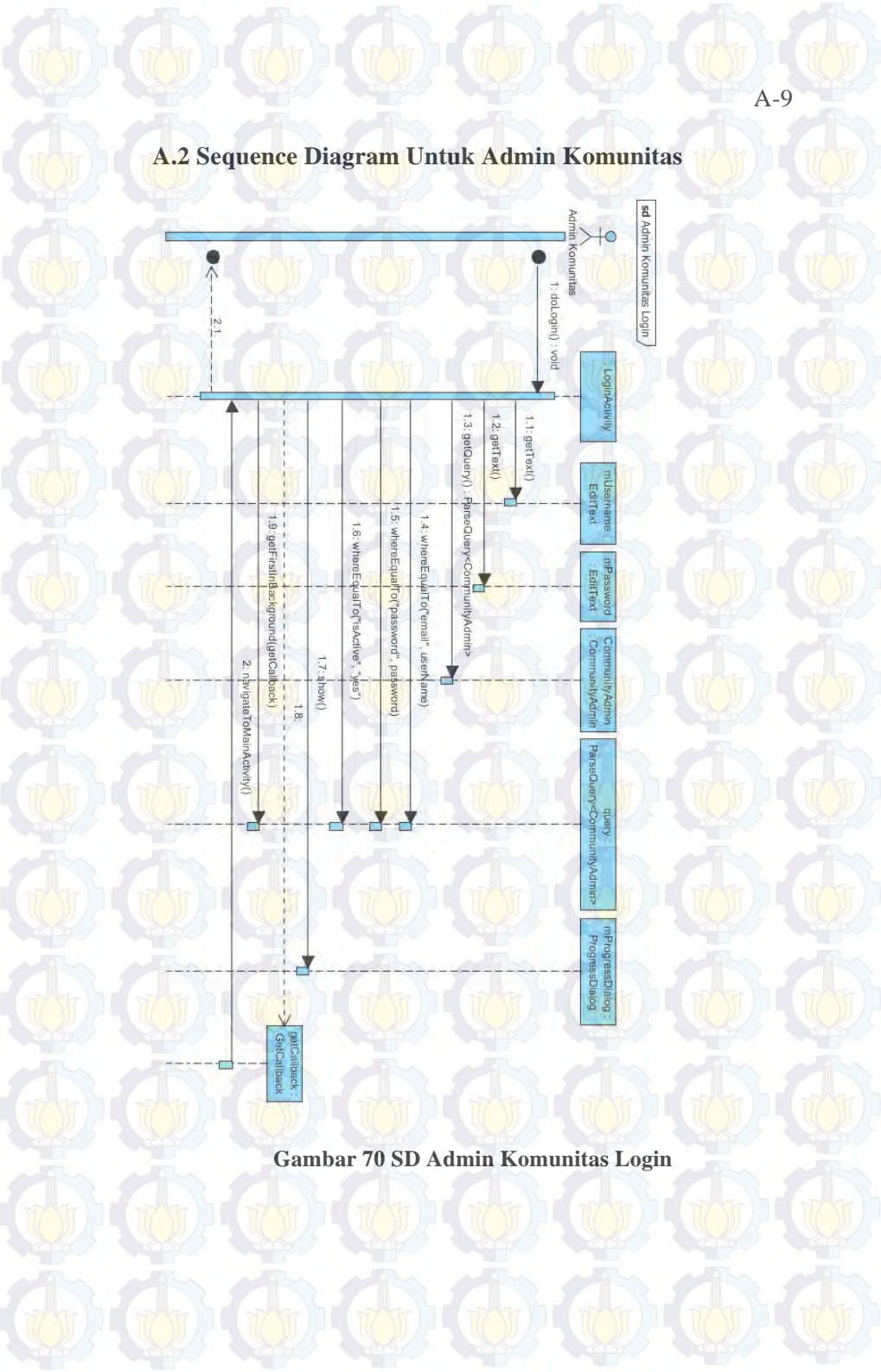
A-9

### A.2 Sequence Diagram Untuk Admin Komunitas

```
sequenceDiagram
    actor Admin as sd Admin Komunitas login
    participant LoginActivity
    participant mUsername as mUsername - EditText
    participant mPassword as mPassword - EditText
    participant CommunityAdmin as CommunityAdmin - CommunityAdmin
    participant mQuery as mQuery - QueryCommunityAdmin
    participant mProgressDialog as mProgressDialog - ProgressDialog
    participant getCallback as getCallback - GetCallback

    Admin->>LoginActivity: 1: doLogin() : void
    activate LoginActivity
    LoginActivity->>mUsername: 1.1: get text()
    deactivate mUsername
    LoginActivity->>mPassword: 1.2: get text()
    deactivate mPassword
    LoginActivity->>CommunityAdmin: 1.3: getQuery() : FireQuery<CommunityAdmin>
    deactivate CommunityAdmin
    CommunityAdmin-->>mQuery: 
    deactivate mQuery
    mQuery->>mProgressDialog: 1.4: whereEqualOf(Email', 'userName')
    deactivate mProgressDialog
    mQuery->>mProgressDialog: 1.5: whereEqualOf(password', 'password')
    deactivate mProgressDialog
    mQuery->>mProgressDialog: 1.6: whereEqualOf(isActive', 'yes')
    deactivate mProgressDialog
    mProgressDialog-->>getCallback: 
    deactivate getCallback
    getCallback->>mProgressDialog: 1.7: show()
    deactivate mProgressDialog
    mProgressDialog->>LoginActivity: 1.8: get finishing background(getCallback)
    deactivate LoginActivity
    LoginActivity->>Admin: 2: navigateToMainActivity()
    deactivate Admin
    Admin-->>Admin: <--> 2.1
```

Gambar 70 SD Admin Komunitas Login



A-9

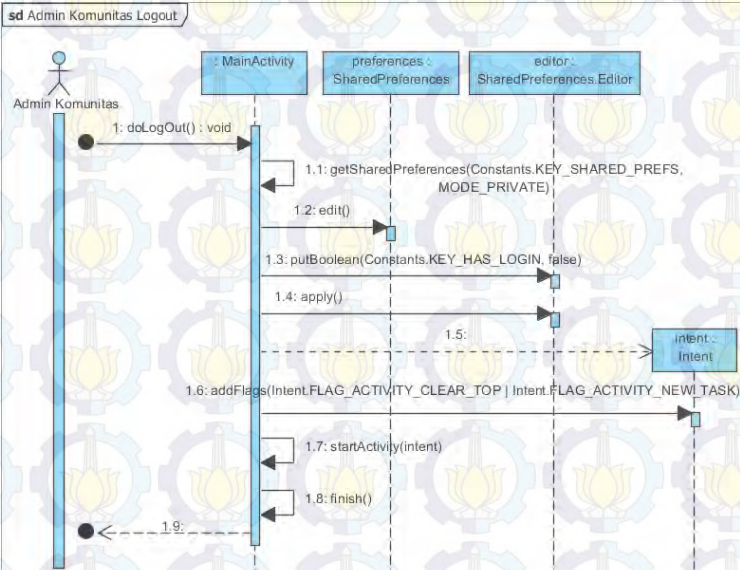
### A.2 Sequence Diagram Untuk Admin Komunitas

```
sequenceDiagram
    actor Admin as sd Admin Komunitas login
    participant LoginActivity
    participant mUsername as mUsername - EditText
    participant mPassword as mPassword - EditText
    participant CommunityAdmin as CommunityAdmin - CommunityAdmin
    participant mQuery as mQuery - QueryCommunityAdmin
    participant mProgressDialog as mProgressDialog - ProgressDialog
    participant getCallback as getCallback - GetCallback

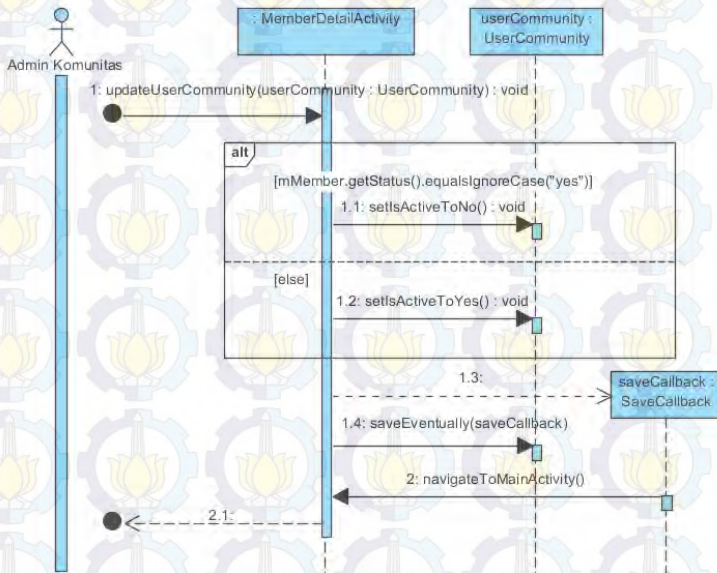
    Admin->>LoginActivity: 1: doLogin() : void
    activate LoginActivity
    LoginActivity->>mUsername: 1.1: get text()
    deactivate mUsername
    LoginActivity->>mPassword: 1.2: get text()
    deactivate mPassword
    LoginActivity->>CommunityAdmin: 1.3: getQuery() : FireQuery<CommunityAdmin>
    deactivate CommunityAdmin
    CommunityAdmin-->>mQuery: query
    deactivate mQuery
    mQuery->>mProgressDialog: 1.4: whereEqualOf(Email', 'userName)'
    deactivate mProgressDialog
    mQuery->>mProgressDialog: 1.5: whereEqualOf(password', 'password)'
    deactivate mProgressDialog
    mQuery->>mProgressDialog: 1.6: whereEqualOf(isactive', 'yes')
    deactivate mProgressDialog
    mProgressDialog-->>getCallback: 
    deactivate getCallback
    getCallback->>CommunityAdmin: 1.7: show()
    deactivate CommunityAdmin
    getCallback->>LoginActivity: 1.8: get finishing background(getCallback)
    deactivate LoginActivity
    getCallback->>LoginActivity: 2: navigateToMainActivity()
    deactivate LoginActivity
    LoginActivity-->>Admin: 2.1
    deactivate LoginActivity
```

Gambar 70 SD Admin Komunitas Login



**Gambar 71 SD Admin Komunitas Logout**

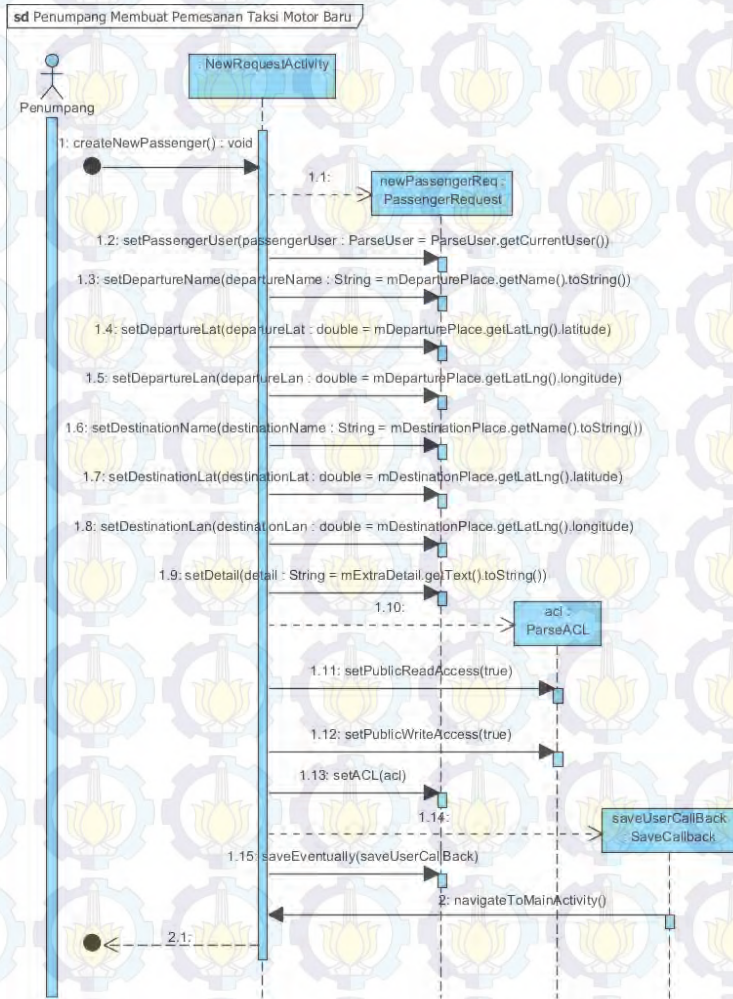
## sd Admin Komunitas Mengubah Status Verifikasi Member



**Gambar 72 SD Admin Komunitas Mengubah Status Verifikasi Member**

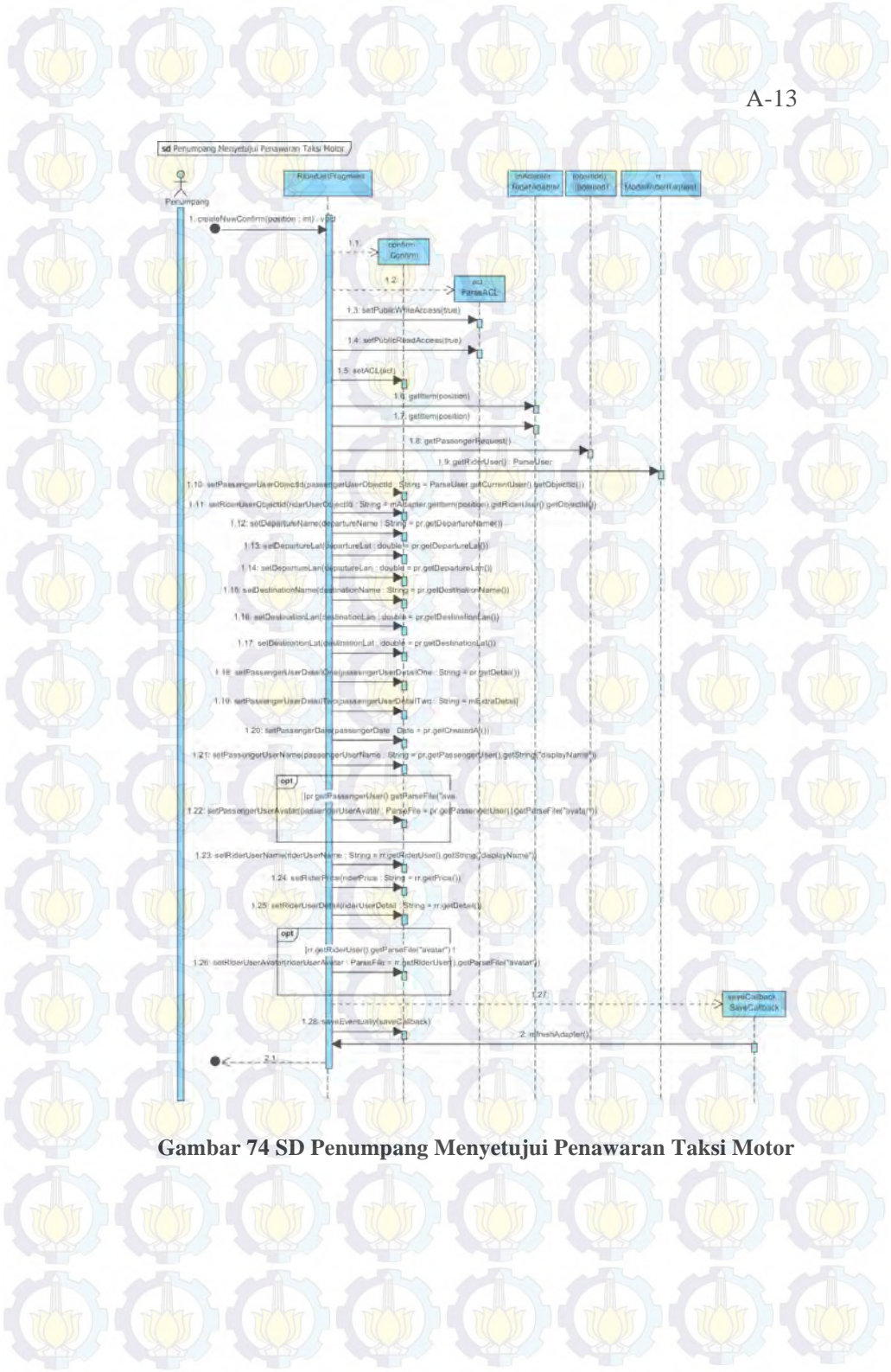


### A.3 Sequence Diagram Untuk Penumpang



**Gambar 73 SD Penumpang Membuat Pemesanan Taksi Motor Baru**





A-13

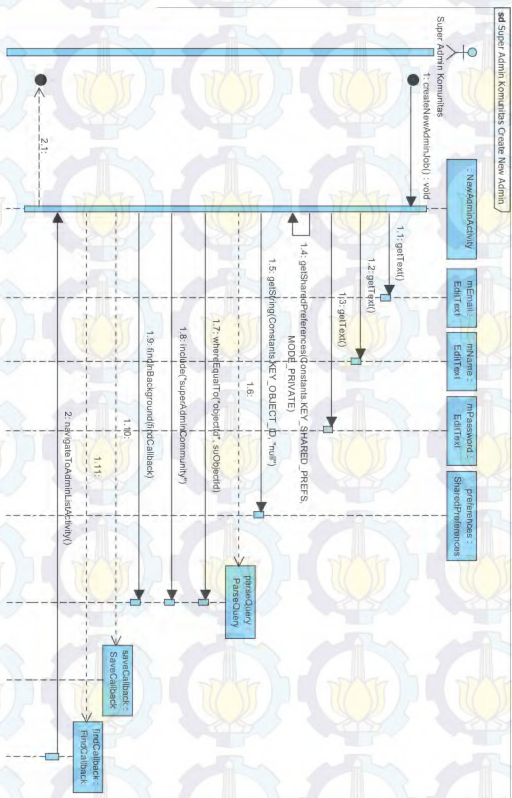
**sd Penumpang Menyetujui Penawaran Taksi Motor**

Participants: Penumpang, RideOfferResponse, mAdapter, RideOffer, RideOfferDetail, mCallback

Sequence of Messages:

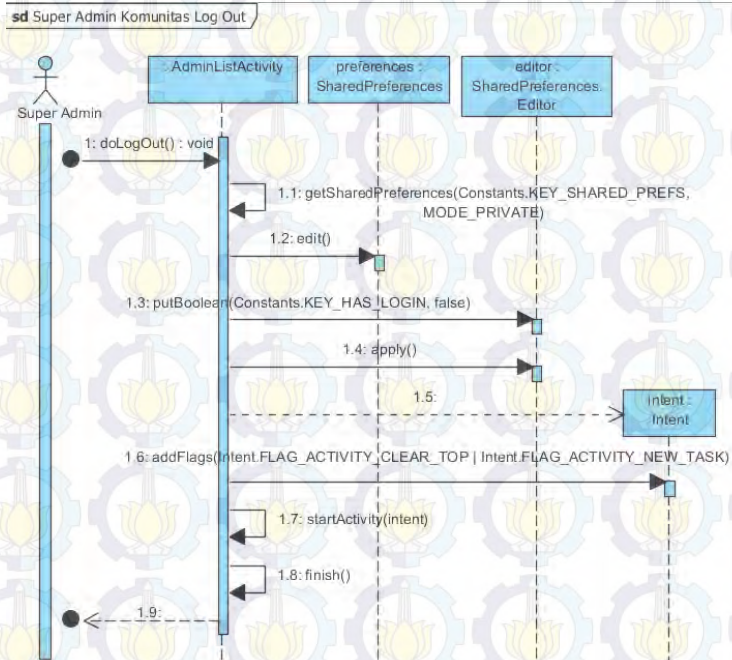
- Penumpang → RideOfferResponse: 1. createNewConfirmation(position: int): int
- RideOfferResponse → RideOffer: 1.1. getOfferId() (return: int)
- RideOfferResponse → RideOffer: 1.2. setPublicWriteAccess(true)
- RideOfferResponse → RideOffer: 1.3. setPublicReadAccess(true)
- RideOfferResponse → RideOffer: 1.4. setACL(id)
- RideOfferResponse → RideOffer: 1.5. getOfferId()
- RideOfferResponse → RideOffer: 1.6. getOfferId()
- RideOfferResponse → RideOffer: 1.7. getOfferId()
- RideOfferResponse → RideOffer: 1.8. getOfferId()
- RideOfferResponse → RideOffer: 1.9. getOfferId()
- RideOfferResponse → RideOffer: 1.10. setPassengerUserObjectid(passengerUserObjectid: String = pr.getCurrentUserId().getOfferId())
- RideOfferResponse → RideOffer: 1.11. setRideOfferObjectid(rideOfferObjectid: String = mAdapter.getOfferId().getOfferId())
- RideOfferResponse → RideOffer: 1.12. setDepartureName(departureName: String = pr.getDepartureName())
- RideOfferResponse → RideOffer: 1.13. setDepartureLat(departureLat: double = pr.getDepartureLat())
- RideOfferResponse → RideOffer: 1.14. setDepartureLon(departureLon: double = pr.getDepartureLon())
- RideOfferResponse → RideOffer: 1.15. setDestinationName(destinationName: String = pr.getDestinationName())
- RideOfferResponse → RideOffer: 1.16. setDestinationLat(destinationLat: double = pr.getDestinationLat())
- RideOfferResponse → RideOffer: 1.17. setDestinationLon(destinationLon: double = pr.getDestinationLon())
- RideOfferResponse → RideOffer: 1.18. setPassengerUserDetailOne(passengerUserDetailOne: String = pr.getDetailOne())
- RideOfferResponse → RideOffer: 1.19. setPassengerUserDetailTwo(passengerUserDetailTwo: String = pr.getDetailTwo())
- RideOfferResponse → RideOffer: 1.20. setPassengerDate(passengerDate: Date = pr.getCreateAt())
- RideOfferResponse → RideOffer: 1.21. setPassengerUserName(passengerUserName: String = pr.getPassengerUser().getString("displayName"))
- RideOfferResponse → RideOffer: 1.22. setPassengerUserAvatar(passengerUserAvatar: Profile = pr.getPassengerUser().getProfile("avatar"))
- RideOfferResponse → RideOffer: 1.23. setRideOfferName(rideOfferName: String = m.getRideOffer().getString("displayName"))
- RideOfferResponse → RideOffer: 1.24. setRideOfferPrice(rideOfferPrice: String = m.getRideOffer().getString("price"))
- RideOfferResponse → RideOffer: 1.25. setRideOfferDetail(rideOfferDetail: String = m.getRideOffer().getString("detail"))
- RideOfferResponse → RideOffer: 1.26. setRideOfferAvatar(rideOfferAvatar: Profile = m.getRideOffer().getProfile("avatar"))
- RideOfferResponse → RideOffer: 1.27. setRideOfferCallback(rideOfferCallback: mCallback)
- RideOfferResponse → RideOffer: 2. mAdapter()
- RideOfferResponse → RideOffer: 2.1. mCallback()

# A.4 Sequence Diagram Super Admin Komunitas



Gambar 75 SD Super Admin Komunitas Create New Admin





**Gambar 76 SD Super Admin Komunitas Log Out**





Gambar 77 SD Super Admin Komunitas Login

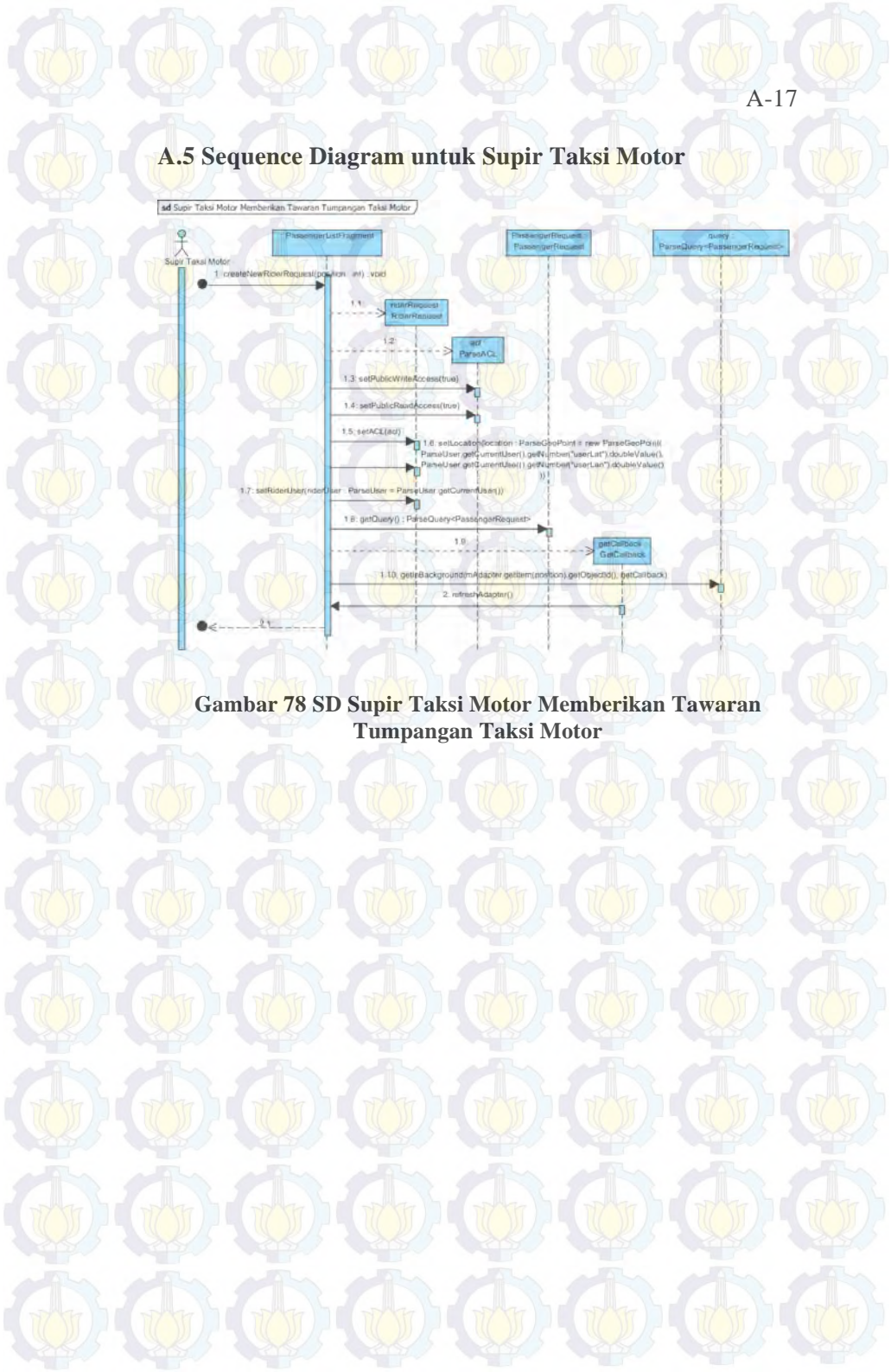
A-17

### A.5 Sequence Diagram untuk Supir Taksi Motor

```
sequenceDiagram
    actor Supir as Supir Taksi Motor
    participant PassengerListFragment
    participant PassengerRequest
    participant ParseACL
    participant ParseGeoPoint as ParseGeoPoint
    participant ParseUser
    participant ParseQuery
    participant ParseInBackground as ParseInBackground

    Supir->>PassengerListFragment: 1. createNewRideRequest(location, init) void
    activate PassengerListFragment
    PassengerListFragment->>PassengerRequest: 1.1.
    activate PassengerRequest
    PassengerRequest->>ParseACL: 1.2.
    activate ParseACL
    ParseACL->>ParseGeoPoint: 1.3. setPublicWriteAccess(true)
    activate ParseGeoPoint
    ParseGeoPoint->>ParseUser: 1.4. setPublicReadAccess(true)
    activate ParseUser
    ParseUser->>ParseQuery: 1.5. setACL(aud)
    activate ParseQuery
    ParseQuery->>ParseInBackground: 1.6. setLocation(location : ParseGeoPoint, new ParseGeoPoint(
    ParseUser.getCurrentUser().getInBackground().doubleValue(1),
    ParseUser.getCurrentUser().getInBackground().doubleValue(2))
    activate ParseInBackground
    ParseInBackground->>ParseQuery: 1.7. self.setUserInBackground(ParseUser.getCurrentUser())
    activate ParseQuery
    ParseQuery->>ParseInBackground: 1.8. getQuery() : ParseQuery<PassengerRequest>
    activate ParseInBackground
    ParseInBackground->>ParseQuery: 1.9.
    activate ParseQuery
    ParseQuery->>ParseInBackground: 1.10. getInBackground(adapter.getLimit(), adapter.getOffset(), adapter.getCallback())
    activate ParseInBackground
    ParseInBackground->>PassengerListFragment: 2. refreshAdapter()
    deactivate ParseInBackground
    ParseQuery->>Supir: 3.1.
    deactivate ParseQuery
    deactivate ParseInBackground
    deactivate ParseUser
    deactivate ParseGeoPoint
    deactivate ParseACL
    deactivate PassengerRequest
    deactivate PassengerListFragment
```

Gambar 78 SD Supir Taksi Motor Memberikan Tawaran Tumpangan Taksi Motor



A-17

### A.5 Sequence Diagram untuk Supir Taksi Motor

```
sequenceDiagram
    actor Supir Taksi Motor
    participant PassengerListFragment
    participant PassengerRequest
    participant ParseACL
    participant ParseGeoPoint
    participant ParseUser
    participant ParseQuery
    participant ParseInBackground

    Supir Taksi Motor->>PassengerListFragment: 1. createNewRideRequest(location, init) void
    activate PassengerListFragment
    PassengerListFragment->>PassengerRequest: 1.1.
    activate PassengerRequest
    PassengerRequest->>ParseACL: 1.2.
    activate ParseACL
    ParseACL->>ParseGeoPoint: 1.3. setPublicWriteAccess(true)
    activate ParseGeoPoint
    ParseGeoPoint->>ParseUser: 1.4. setPublicReadAccess(true)
    activate ParseUser
    ParseUser->>ParseQuery: 1.5. setACL(aud)
    activate ParseQuery
    ParseQuery->>ParseInBackground: 1.6. setLocation(location : ParseGeoPoint = new ParseGeoPoint(
    ParseUser.getCurrentUser().getLatitude(), ParseUser.getCurrentUser().getLongitude())
    ParseQuery->>ParseInBackground: 1.7. self.setUser(userId : ParseUser = ParseUser.getCurrentUser())
    ParseQuery->>ParseInBackground: 1.8. getQuery() : ParseQuery<PassengerRequest>
    ParseInBackground->>ParseQuery: 1.9.
    ParseInBackground->>ParseQuery: 1.10. getInBackground(adapter.getLimit(), adapter.getOffset(), adapter.getCallback())
    ParseQuery->>Supir Taksi Motor: 2. refreshAdapter()
    deactivate ParseQuery
    deactivate ParseInBackground
    deactivate ParseUser
    deactivate ParseGeoPoint
    deactivate ParseACL
    deactivate PassengerRequest
    deactivate PassengerListFragment
```

Gambar 78 SD Supir Taksi Motor Memberikan Tawaran Tumpangan Taksi Motor



*(halaman ini sengaja dikosongkan)*



## LAMPIRAN B CLASS DIAGRAM

### B.1 Class Diagram Untuk Admin Komunitas

<b>CommunityAdminApplication</b> (com::aldoapps::ojekfinderadmin)
+onCreate() : void

Gambar 79 CD CommunityAdminApplication

#### B.1.1 Adapter

<b>MemberItemViewAdapter</b> (com::aldoapps::ojekfinderadmin)
-mMemberList : Member
+MemberItemViewAdapter(members : ArrayList<Member>)
+onCreateViewHolder(parent : ViewGroup, viewType : int) : ViewHolder
+onBindViewHolder(holder : ViewHolder, position : int) : void
+getItemCount() : int

Gambar 80 CD MemberItemViewAdapter

B-2

### B.1.2 Model



Gambar 81 CD CommunityAdmin



Gambar 82 CD Member



<b>ModelUserCommunity</b> (com::aldoapps::ojekfinderadmin::model)
-objectId : String
-userObjectId : String
-communityObjectId : String
-isActive : String
+ModelUserCommunity()

Gambar 83 CD ModelUserCommunity

<b>UserC</b> (com::aldoapps::ojekfinderadmin::model)
+IS_PASSENGER : String = "isPassenger"
+IS_RIDER : String = "isRider"
+ACTIVATION_STATUS : String = "activationStatus"
+AVATAR : String = "avatar"

Gambar 84 CD UserC

<b>UserCommunity</b> (com::aldoapps::ojekfinderadmin::model)
-objectId : String
-communityObjectId : String
-isActive : String
+UserCommunity()
+isCurrentlyActive() : boolean
+haveCommunity() : boolean
+setIsActiveToNo() : void
+setIsActiveToYes() : void
+getQuery() : ParseQuery<UserCommunity>

Gambar 85 CD UserCommunity



### B.1.3 User Interface

<b>LoginActivity</b> (com::aldoapps::ojekfinderadmin)
-TAG : String = LoginActivity.class.getSimpleName()
-mUsername : EditText
-mPassword : EditText
-mLoginBtn : Button
-mProgressDialog : ProgressDialog
#onCreate(savedInstanceState : Bundle) : void
-doLogin() : void
-saveCommunityAdminToLocalStorage(communityAdmin : CommunityAdmin) : void
-navigateToMainActivity() : void
#onPause() : void

**Gambar 86 CD LoginActivity**

<b>MainActivity</b> (com::aldoapps::ojekfinderadmin)
-TAG : String = MainActivity.class.getSimpleName()
-mListView : RecyclerView
-mMembers : Member = new ArrayList<>()
-mUserComms : ModelUserCommunity = new ArrayList<>()
-mAdapter : MemberItemViewAdapter
-mRefresh : SwipeRefreshLayout
#onCreate(savedInstanceState : Bundle) : void
-loadParseUsers() : void
-loadUserCommunity(communityAdmin : CommunityAdmin) : void
-filterActualUsers() : void
+onBackPressed() : void
+onNavigationItemSelected(item : MenuItem) : boolean
-doLogout() : void

**Gambar 87 CD MainActivity**

<b>MemberDetailFragment</b> (com::aldoapps::ojekfinderadmin)
+onCreate(savedInstanceState : Bundle) : void

**Gambar 88 CD MemberDetailFragment**

<b>ViewHolder</b>
(com::aldoapps::ojekfinderadmin::MemberItemViewAdapter)
+mView : View
+mAvatar : LetterImageView
+mName : TextView
+mStatus : TextView
+mRating : RatingBar
+ViewHolder(itemView : View)

Gambar 89 CD ViewHolder

### B.1.4 Utility Class

<b>Constants</b>
(com::aldoapps::ojekfinderadmin)
+KEY_HAS_LOGIN : String = "KEY_HAS_LOGIN"
+KEY_SHARED_PREFS : String = "KEY_SHARED_PREFS"
+KEY_ADMIN_OBJECT_ID : String = "KEY_ADMIN_OBJECT_ID"

Gambar 90 CD Constants

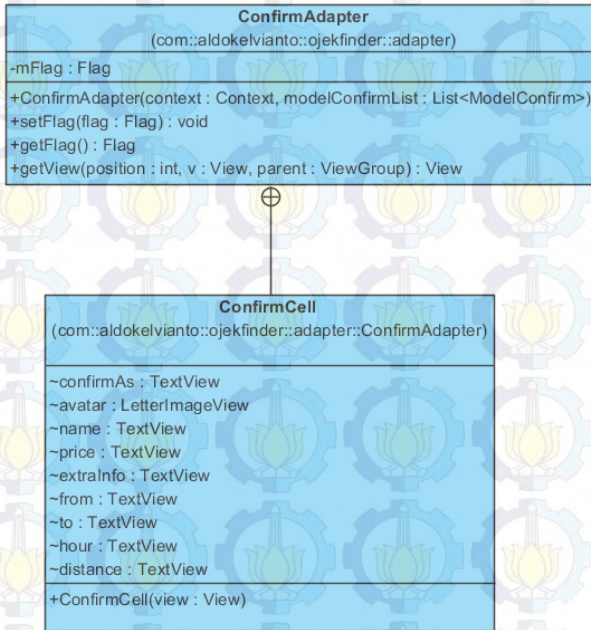
## B.2 Class Diagram Untuk Ojek Finder

<b>OjekFinderApplication</b>
(com::aldokelvianto::ojekfinder)
+onCreate() : void
-registerCurrentInstallation() : void
-registerApiKey() : void
-registerSubclasses() : void
-initializeLocalDatastore() : void

Gambar 91 CD OjekFinderApplication



## B.2.1 Adapter

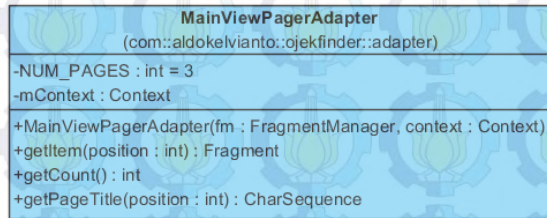


**Gambar 92 CD ConfirmAdapter**

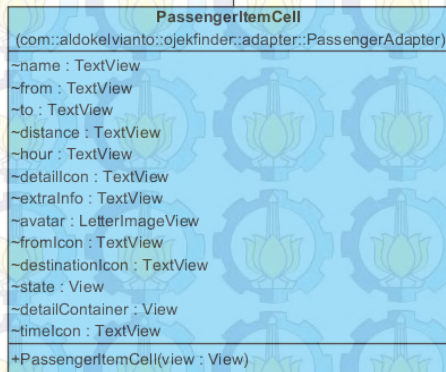
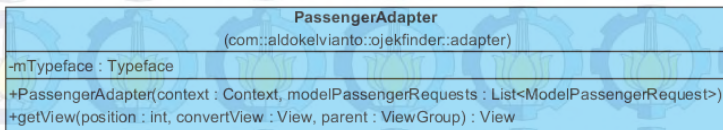


**Gambar 93 CD LandingViewPagerAdapter**





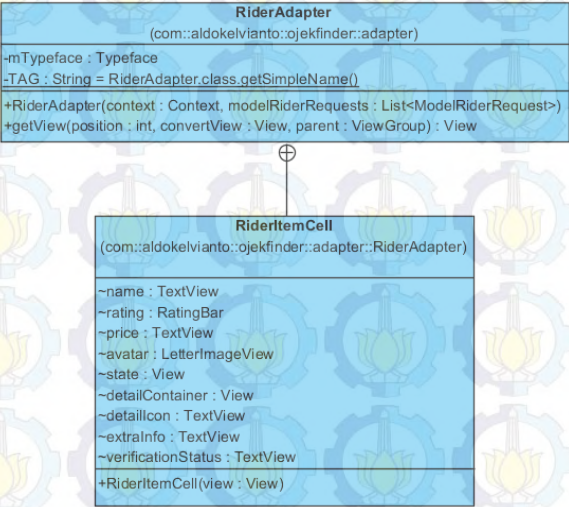
Gambar 94 CD MainViewPagerAdapter



Gambar 95 CD PassengerAdapter



**Gambar 96 CD ReviewRatingAdapter**



**Gambar 97 CD RiderAdapter**



## B.2.2 Model

<b>Community</b> (com::aldokelvianto::ojekfinder::model)
-name : String
+getQuery() : ParseQuery<Community>

**Gambar 98 CD Community**

<b>Confirm</b> (com::aldokelvianto::ojekfinder::model)
-riderUserObjectld : String
-passengerUserObjectld : String
-departureLat : double
-departureLan : double
-departureName : String
-destinationLat : double
-destinationLan : double
-destinationName : String
-riderUserName : String
-riderUserAvatar : ParseFile
-riderUserDetail : String
-passengerUserName : String
-passengerUserAvatar : ParseFile
-passengerUserDetailOne : String
-passengerUserDetailTwo : String
-riderPrice : String
-passengerDate : Date
+Confirm()
+getQuery() : ParseQuery<Confirm>

**Gambar 99 CD Confirm**



<b>ConfirmFlag</b>
(com::aldokelvianto::ojekfinder::model)
-confirmObjectId : String
-isPassengerFinish : boolean
-isRiderFinish : boolean
-query : ParseQuery<ConfirmFlag>
+ConfirmFlag()
+isPassengerFinish() : boolean
+isRiderFinish() : boolean

Gambar 100 CD ConfirmFlag

<b>Flag</b>
(com::aldokelvianto::ojekfinder::model)
-userId : String
-isPassenger : String
-isRider : String
-isConfirmed : String
+Flag()
+setIsPassengerToYes() : void
+setIsPassengerToNo() : void
+setIsRiderToYes() : void
+setIsRiderToNo() : void
+isCurrentlyARider() : boolean
+isCurrentlyAPassenger() : boolean
+isCurrentlyConfirmed() : boolean
+setIsConfirmedToYes() : void
+setIsConfirmedToNo() : void
+getQuery() : ParseQuery<Flag>

Gambar 101 CD Flag

<b>ModelConfirm</b>
(com::aldokelvianto::ojekfinder::model)
-objectId : String
-riderUserObjectId : String
-passengerUserObjectId : String
-departureLat : double
-departureLan : double
-departureName : String
-destinationLat : double
-destinationLan : double
-destinationName : String
-riderUserName : String
-riderUserAvatar : ParseFile
-riderUserDetail : String
-passengerUserName : String
-passengerUserAvatar : ParseFile
-passengerUserDetailOne : String
-passengerUserDetailTwo : String
-riderPrice : String
-passengerDate : Date
+ModelConfirm()

Gambar 102 CD ModelConfirm

<b>ModelPassengerRequest</b>
(com::aldokelvianto::ojekfinder::model)
-objectId : String
-createdAt : Date
-passengerUser : ParseUser
-departureLat : double
-departureLan : double
-departureName : String
-destinationLat : double
-destinationLan : double
-destinationName : String
-detail : String
-distance : double
-channelUUID : String
+ModelPassengerRequest()

Gambar 103 CD ModelPassengerRequest



ModelReviewRating
(com::aldokelvianto::ojekfinder::model)
-objectId : String
-createdAt : Date
-senderObjectId : String
-targetObjectId : String
-review : String
-rating : float
+ModelReviewRating()

Gambar 104 CD ModelReviewRating

ModelRiderRequest
(com::aldokelvianto::ojekfinder::model)
-objectId : String
-passengerRequest : PassengerRequest
-riderUser : ParseUser
-location : ParseGeoPoint
-price : String
-detail : String
-isRider : String
-rating : float
+ModelRiderRequest()

Gambar 105 CD ModelRiderRequest

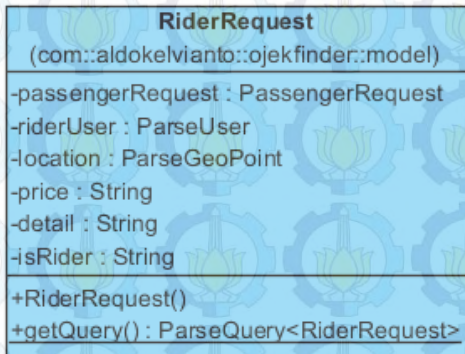


<b>PassengerRequest</b> (com::aldokelvianto::ojekfinder::model)
-passengerUser : ParseUser -departureLat : double -departureLan : double -departureName : String -destinationLat : double -destinationLan : double -destinationName : String -detail : String -distance : double -channelUUID : String
+PassengerRequest() +getQuery() : ParseQuery<PassengerRequest>

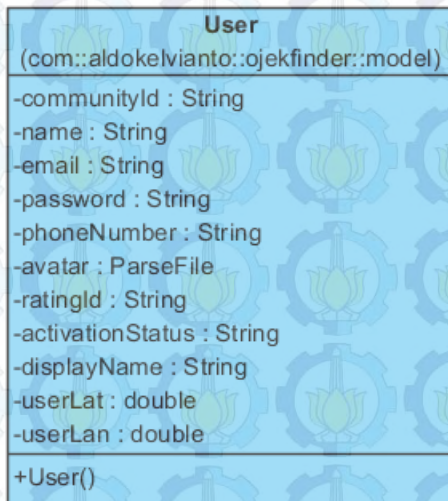
Gambar 106 CD PassengerRequest

<b>ReviewRating</b> (com::aldokelvianto::ojekfinder::model)
-senderObjectId : String -targetObjectId : String -review : String -rating : float
+ReviewRating() +getQuery() : ParseQuery<ReviewRating>

Gambar 107 CD ReviewRating



Gambar 108 CD RiderRequest



Gambar 109 CD User



<b>UserCommunity</b> (com::aldokelvianto::ojekfinder::model)
-userId : String -communityObjectId : String -isActive : String
+UserCommunity() +isCurrentlyActive() : boolean +haveCommunity() : boolean +getQuery() : ParseQuery<UserCommunity> +setIsActiveToNo() : void +setIsActiveToYes() : void

Gambar 110 CD UserCommunity



## B.2.3 User Interface

ConfirmedListFragment (com::aldokelvianto::ojekfinder:ui)
<pre> +TAG : String = ConfirmedListFragment.class.getSimpleName() ~mListView : ListView ~mRefresh : SwipeRefreshLayout -mContext : MainActivity -mAdapter : ConfirmAdapter -mConfirmList : List&lt;ModelConfirm&gt; = new ArrayList&lt;&gt;() -mConfirm : ModelConfirm -mFlag : Flag -mIsPassenger : boolean = false -mIsRider : boolean = false  +ConfirmedListFragment() +onAttach(activity : Activity) : void +newInstance() : ConfirmedListFragment +onResume() : void +onCreateContextMenu(menu : ContextMenu, v : View, menuInfo : ContextMenuInfo) : void +onContextItemSelected(item : MenuItem) : boolean -navigateToLocationDetail(position : int) : void +onCreate(savedInstanceState : Bundle) : void -startLoadConfirms() : void -loadFlagAsFilters() : void -loadConfirms(flag : Flag) : void -convertConfirmList(list : List&lt;Confirm&gt;) : void +onCreateView(inflater : LayoutInflater, container : ViewGroup, savedInstanceState : Bundle) : View +onViewCreated(view : View, savedInstanceState : Bundle) : void -showConfirmationDialog() : void -startInspectingConfirmFlag() : void -findExistingConfirmFlag() : void -checkConfirmFlag(confirmFlag : ConfirmFlag) : void -setUserConfirmFlagToFinish(confirmFlag : ConfirmFlag) : void -startInitiateConfirmFlag(confirmObjectId : String) : void -createInitialConfirmFlag(confirmObjectId : String) : void -deleteConfirm() : void -deleteConfirmFlagAndConfirm() : void -showReviewRatingDialog() : void -sendReviewRating(commentText : String, ratingValue : float) : void -updateOtherUserFlag() : void -updateUserFlag() : void -setEveryFlagToNo(flag : Flag) : void </pre>

Gambar 111 CD ConfirmedListFragment

<b>EditProfileActivity</b> (com::aldokelvianto::ojekfinder::ui)
~mToolBar : Toolbar ~mEmail : TextView ~mDisplayName : EditText ~mPhoneNumber : EditText ~mSaveButton : Button ~mOFDialog : OFDialog
#onCreate(savedInstanceState : Bundle) : void -saveChanges() : void -initToolBar() : void

Gambar 112 CD EditProfileActivity

<b>LandingActivity</b> (com::aldokelvianto::ojekfinder::ui)
+containerId : int = R.id.container
#onCreate(savedInstanceState : Bundle) : void -launchLandingFragment() : void

Gambar 113 CD LandingActivity

<b>LandingFragment</b> (com::aldokelvianto::ojekfinder::ui)
+TAG : String = "landing_fragment" ~mViewPager : ViewPager ~mLoginBtn : Button ~mSignUpBtn : Button ~mDotsLayout : LinearLayout ~mAdapter : LandingViewPagerAdapter ~mDotsCount : int ~mDotsTextView : TextView[]
+newInstance() : LandingFragment +onCreateView(inflater : LayoutInflater, container : ViewGroup, savedInstanceState : Bundle) : View +onResume() : void +onViewCreated(view : View, savedInstanceState : Bundle) : void -setListenersForEachButton() : void -launchLoginFragment() : void -launchSignUpFragment() : void +onPageScrolled(position : int, positionOffset : float, positionOffsetPixels : int) : void +onPageSelected(position : int) : void +onPageScrollStateChanged(state : int) : void

Gambar 114 CD LandingFragment



<b>LocationDetailActivity</b> (com::aldokelvianto::ojekfinder::ui)
-mMap : GoogleMap -mDestination : LatLng -mDeparture : LatLng -mRiderLocation : LatLng +KEY_DESTINATION : String = "key_destination" +KEY_DEPARTURE : String = "key_departure" +KEY_RIDER_LOCATION : String = "key_rider_location" -ZOOM_LEVEL : float = 14f -ZOOM_LEVEL_RIDER : float = 15f -ANIMATE_SPEED : int = 1 -mIsRiderDetail : boolean = false
#onCreate(savedInstanceState : Bundle) : void -initializeToolbar() : void -setUpMapIfNeeded() : void -setUpMapForPassenger() : void -setUpMapForRider() : void

Gambar 115 CD LocationDetailActivity

<b>LoginFragment</b> (com::aldokelvianto::ojekfinder::ui)
~mUserNameTxt : EditText ~mPasswordTxt : EditText ~mLoginBtn : Button +TAG : String = "login_fragment" -mProgressDialog : ProgressDialog
+newInstance() : LoginFragment +onResume() : void +onCreateView(inflater : LayoutInflater, container : ViewGroup, savedInstanceState : Bundle) : View +onViewCreated(view : View, savedInstanceState : Bundle) : void -setLoginListener() : void -doLogin() : void -navigateToMainActivity() : void +onPause() : void

Gambar 116 CD LoginFragment



MainActivity (com:aldokelvianto::ojekfinder::ui)
-CONNECTION FAILURE RESOLUTION REQUEST : int = 9000
-TAG : String = MainActivity.class.getSimpleName()
~mToolBar : Toolbar
~mDrawerLayout : DrawerLayout
~mNavigationView : NavigationView
-mGoogleApiClient : GoogleApiClient
-mDrawerToggle : ActionBarDrawerToggle
-mContainerId : int = R.id.container
-mHasLoggedIn : boolean
-mNavHeaderLayout : RelativeLayout
-mLocation : Location
#onCreate(savedInstanceState : Bundle) : void
#onStart() : void
#onPause() : void
#onResume() : void
-initNavigationView() : void
-navigateToRequestCommunity() : void
-navigateToSetting() : void
-showAboutDialog() : void
-logout() : void
-initToolBar() : void
-initDrawer() : void
+onCreateOptionsMenu(menu : Menu) : boolean
+onOptionsItemSelected(item : MenuItem) : boolean
+launchMapsActivity() : void
-launchMainFragment() : void
+onBackPressed() : void
+onConnected(bundle : Bundle) : void
-saveCurrentUserLocation() : void
+onConnectionSuspended(i : int) : void
+onConnectionFailed(connectionResult : ConnectionResult) : void

Gambar 117 CD MainActivity

<b>MainFragment</b> (com::aldokelvianto::ojekfinder:ui)
<pre>+TAG : String = MainFragment.class.getSimpleName() ~mViewPager : ViewPager ~mTabLayout : TabLayout ~mFab : FloatingActionButton -mMainViewPagerAdapter : MainViewPagerAdapter -mContext : MainActivity -mContainerId : int = R.id.container ~mProgressDialog : ProgressDialog</pre>
<pre>+newInstance() : MainFragment +onAttach(activity : Activity) : void +onCreate(savedInstanceState : Bundle) : void +onResume() : void +onCreateView(inflater : LayoutInflater, container : ViewGroup, savedInstanceState : Bundle) : View +onViewCreated(view : View, savedInstanceState : Bundle) : void -checkUserFlag() : void -startNewRequestActivity() : void</pre>

Gambar 118 CD MainFragment

<b>MapsActivity</b> (com::aldokelvianto::ojekfinder:ui)
<pre>-ZOOM_LEVEL : float = 13f -ANIMATE_SPEED : int = 1 -TAG : String = MapsActivity.class.getSimpleName() -mMap : GoogleMap -mMarkers : List&lt;MarkerOptions&gt; = new ArrayList&lt;&gt;() -mLat : double -mLan : double</pre>
<pre>#onCreate(savedInstanceState : Bundle) : void -startLoadAllUserLocation() : void -loadAllUserLocation(list : List&lt;ParseUser&gt;) : void -updateMap() : void #onResume() : void -setupMapIfNeeded() : void -setupMap() : void</pre>

Gambar 119 CD MapsActivity



```
NewRequestActivity  
(com::aldokelvianto::ojekfinder::ui)  
  
+TAG : String = NewRequestActivity.class.getSimpleName()  
~mToolbar : Toolbar  
~mFrom : TextView  
~mFromIcon : TextView  
~mDestinationIcon : TextView  
~mDestination : TextView  
~mExtraDetail : EditText  
~mSendRequest : Button  
~mDeparturePlace : Place  
~mDestinationPlace : Place  
  
#onCreate(savedInstanceState : Bundle) : void  
~createNewPassenger() : void  
~updateUserFlag() : void  
~savePassengerFlagToYes(flag : Flag) : void  
~initToolbar() : void  
+onActivityResult(requestCode : int, resultCode : int, data : Intent) : void  
~startFromPicker() : void  
~startDestinationPicker() : void
```

**Gambar 120 CD NewRequestActivity**



```

PassengerListFragment
(com::aldokelvianto::ojekfinder:ui)

-TAG : String = PassengerListFragment.class.getSimpleName()
-MAX_KM : double = 7
~mListView : ListView
~mRefresh : SwipeRefreshLayout
-mAdapter : PassengerAdapter
-mContext : MainActivity
-mSelectedPrice : String
-mDetail : String
-mOFDialog : OFDialog
-mDeletedRequest : PassengerRequest
-mPassengerRequestList : List<ModelPassengerRequest> = new ArrayList<>()

+newInstance() : PassengerListFragment
+PassengerListFragment()
+onAttach(activity : Activity) : void
+onResume() : void
+onCreate(savedInstanceState : Bundle) : void
-startLoadPassengerRequest() : void
-loadPassengerRequest() : void
-filterPassengerRequest(list : List<PassengerRequest>) : void
+onCreateView(inflater : LayoutInflater, container : ViewGroup, savedInstanceState : Bundle) : View
+onViewCreated(view : View, savedInstanceState : Bundle) : void
-checkUserAvailability(position : int) : void
-checkIfUserHasCommunity(position : int) : void
-showCancelRequestDialog(position : int) : void
-deletePassengerRequest(pr : ModelPassengerRequest) : void
-deleteRiderRequest(riderRequest : RiderRequest) : void
-updateRiderFlag(riderRequest : RiderRequest) : void
-deleteRiderRequestAndUpdateTheirFlag() : void
-updateUserAvailability() : void
-updateUserFlag(flag : Flag) : void
+onCreateContextMenu(menu : ContextMenu, v : View, menuInfo : ContextMenuInfo) : void
+onContextItemSelected(item : MenuItem) : boolean
-navigateToLocationDetail(position : int) : void
-showAnswerRequestDialog(position : int) : void
-createNewRiderRequest(position : int) : void
-updateRiderAvailability() : void

```

**Gambar 121 CD PassengerListFragment**

```

ProfileActivity
(comm::aldokelvianto::ojekfinder::ui)

-IMAGE_TYPE : String = "image/*"
-UPLOAD_FROM_GALLERY : int = 17
-UPLOAD_FROM_CAMERA : int = 18
-REQUEST_CROP_PICTURE : int = 19
-EDIT_PROFILE : int = 20
-TAG : String = ProfileActivity.class.getSimpleName()
-outputFile : File
~mToolbar : Toolbar
~mComment : TextView
~mProfileAvatar : LetterImageView
~mProfileCover : ImageView
~mEmail : TextView
~mAvatarGallery : View
~mChangeBtn : ImageView
~mDisplayName : TextView
~mPhoneNumber : TextView
~mChevronRight : TextView
~mCommentContainer : View
~mRating : RatingBar
~mTotalReviewers : TextView
~mTypeface : Typeface
~mOFDDialog : OFDDialog
~mRatingValue : float

#onCreate(savedInstanceState : Bundle) : void
-loadReviewRatings() : void
-filterRating(list : List<ReviewRating>) : void
-setImageFromParseFile() : void
-handleAvatarUpload() : void
-initToolbar() : void
#onActivityResult(requestCode : int, resultCode : int, data : Intent) : void
-updateProfile() : void
-loadImageFromCamera(data : Intent) : void
-loadImageFromGallery(data : Intent) : void
-uploadAvatar(bitmap : Bitmap) : void
-handleCropImage(data : Intent) : void
-beginCrop(source : Uri) : void
-beginCrop(bitmap : Bitmap) : void

```

Gambar 122 CD ProfileActivity



<b>RequestCommunityActivity</b> (com::aldokelvianto::ojekfinder::ui)
- TAG : String = RequestCommunityActivity.class.getSimpleName() ~mRequestBtn : Button ~mToolbar : Toolbar ~mCommunityName : AutoCompleteTextView ~mRemoveBtn : Button ~mCommunityList : List<String> = new ArrayList<>() ~mAdapter : ArrayAdapter<String> ~mOFDialog : OFDialog
#onCreate(savedInstanceState : Bundle) : void ~removeExistingCommunity() : void ~registerToCommunity() : void ~saveUserCommunity(community : Community) : void ~loadCommunities() : void ~loadUserCommunity() : void ~initToolbar() : void

Gambar 123 CD RequestCommunityActivity

<b>ReviewRatingActivity</b> (com::aldokelvianto::ojekfinder::ui)
- TAG : String = ReviewRatingActivity.class.getSimpleName() ~mListView : ListView ~mToolbar : Toolbar ~mRefresh : SwipeRefreshLayout ~mAdapter : ReviewRatingAdapter ~mReviewRatings : List<ModelReviewRating> = new ArrayList<>() ~mOFDialog : OFDialog
#onCreate(savedInstanceState : Bundle) : void ~loadReviewRatings() : void ~convertReviewRating(list : List<ReviewRating>) : void ~initToolbar() : void

Gambar 124 CD ReviewRatingActivity



SignUpFragment (com::aldokelvianto::ojekfinder::ui)
<pre> ~mPasswordTxt : EditText ~mPasswordAgainTxt : EditText ~mUserNameTxt : EditText ~mSignUpBtn : Button ~mHintTextView : TextView ~mProgressDialog : ProgressDialog ~mTextWatcher : TextWatcher +TAG : String = "sign_in_fragment" +newInstance() : SignUpFragment +onResume() : void +onCreateView(inflater : LayoutInflater, container : ViewGroup, savedInstanceState : Bundle) : View +onViewCreated(view : View, savedInstanceState : Bundle) : void ~createNewUserJob() : void ~createUserFlag() : void ~navigateToMainActivity() : void +onPause() : void </pre>

Gambar 125 CD SignUpFragment

WalkthroughOneFragment (com::aldokelvianto::ojekfinder::ui)
<pre> +newInstance() : WalkthroughOneFragment +onCreateView(inflater : LayoutInflater, container : ViewGroup, savedInstanceState : Bundle) : View </pre>

Gambar 126 CD WalkthroughOneFragment

WalkthroughTwoFragment (com::aldokelvianto::ojekfinder::ui)
<pre> +newInstance() : WalkthroughTwoFragment +onCreateView(inflater : LayoutInflater, container : ViewGroup, savedInstanceState : Bundle) : View </pre>

Gambar 127 CD WalkthroughTwoFragment

## B.2.4. Utility Class

BlankFragment (com::aldokelvianto::ojekfinder::util)
<pre> +newInstance() : BlankFragment </pre>

Gambar 128 CD BlankFragment

<b>CircleTransform</b> (com::aldokelvianto::ojekfinder::util)
<b>+transform(source : Bitmap) : Bitmap</b>
<b>+key() : String</b>

Gambar 129 CD CircleTransform

<b>Constants</b> (com::aldokelvianto::ojekfinder::util)
<b>+MY_SHARED_PREFS : String = "com.aldoapps.ojekfinder.OJEK_FINDER_SHARED_PREFS_KEY"</b>
<b>+KEY_HAS_LOGGED_IN : String = "KEY_HAS_LOGGED_IN"</b>
<b>+LOCATION_FROM_PICKER_REQUEST : int = 0</b>
<b>+LOCATION_DESTINATION_PICKER_REQUEST : int = 1</b>
<b>+ICON_FONT_PATH : String = "fonts/material-icon-font.ttf"</b>
<b>+BLUR_SCALE : float = 2f</b>
<b>+BITMAP_SAMPLE : int = 8</b>
<b>+BITMAP_MINIMUM_SAMPLE : int = 1</b>
<b>+BLUR_RADIUS : int = 8</b>

Gambar 130 CD Constants

<b>DrawableUtil</b> (com::aldokelvianto::ojekfinder::util)
<b>+drawableToBitmap(drawable : Drawable) : Bitmap</b>

Gambar 131 CD DrawableUtil

<b>Earth</b> (com::aldokelvianto::ojekfinder::util)
<b>+distance(lat1 : double, lat2 : double, lon1 : double, lon2 : double, el1 : double, el2 : double) : double</b>
<b>+distance(lat1 : double, lat2 : double, lon1 : double, lon2 : double) : double</b>
<b>+distanceInKm(lat1 : double, lat2 : double, lon1 : double, lon2 : double) : double</b>

Gambar 132 CD Earth



```

FontUtil
(com::aldokelvianto::ojekfinder::util)

-fontMap : Map<String, Character> = new HashMap<>()

+FontUtil()

```

Gambar 133 CD FontUtil

```

GeneratePicture
(com::aldokelvianto::ojekfinder::util)

~MAX_WIDTH : int = 600
~MAX_HEIGHT : int = 800
~uri : Uri
~resolver : ContentResolver
~path : String
~orientation : Matrix
~storedHeight : int
~storedWidth : int

+GeneratePicture(uri : Uri, resolver : ContentResolver)
~getInformation() : boolean
~getInformationFromMediaDatabase() : boolean
~getInformationFromFileSystem() : boolean
~getStoredDimensions() : boolean
+getBitmap() : Bitmap

```

Gambar 134 CD GeneratePicture

```

OFDialog
(com::aldokelvianto::ojekfinder::util)

+OFDialog(context : Context)

```

Gambar 135 CD OFDialog

```

PictureUtil
(com::aldokelvianto::ojekfinder::util)

+scaleBitmap(imageView : ImageView, imagePath : String) : Bitmap

```

Gambar 136 CD PictureUtil



<b>RandomStringUUID</b> (com::aldokelvianto::ojekfinder::util) <hr/> +generateString(rng : Random, characters : String, length : int) : String <hr/> +generateString(rng : Random) : String
--

Gambar 137 CD RandomStringUUID

<b>RenderScriptBlur</b> (com::aldokelvianto::ojekfinder::util) <hr/> +blurFromRenderScript(context : Context, originalBitmap : Bitmap, blurAmount : float) : Bitmap
---

Gambar 138 CD RenderScriptBlur

<b>SoftKeyboardUtil</b> (com::aldokelvianto::ojekfinder::util) <hr/> +hideKeyboard(activity : Activity) : void <hr/> +hideKeyboard(context : Context, view : View) : void
--

Gambar 139 CD SoftKeyboardUtil

<b>StackBlur</b> (com::aldokelvianto::ojekfinder::util) <hr/> +fastblur(sentBitmap : Bitmap, scale : float, radius : int) : Bitmap
--

Gambar 140 CD StackBlur

<b>Toaster</b> (com::aldokelvianto::ojekfinder::util)
+fromFragment(fragment : Fragment, message : String) : void
+fromFragmentDeleteRequestSuccess(fragment : Fragment) : void
+fromFragmentUpdateFlagSuccess(fragment : Fragment) : void
+fromFragmentSendRequestSuccess(fragment : Fragment) : void
+fromFragmentSendRequestFail(fragment : Fragment) : void
+fromActivity(activity : Activity, message : String) : void
+fromActivityDeleteRequestSuccess(activity : Activity) : void
+fromActivityUpdateFlagSuccess(activity : Activity) : void
+fromActivitySendRequestSuccess(fragment : Fragment) : void
+fromActivitySendRequestFail(fragment : Fragment) : void
+fromFragmentSendReviewSuccess(fragment : ConfirmedListFragment) : void

Gambar 141 CD Toaster

<b>TypefaceCache</b> (com::aldokelvianto::ojekfinder::util)
-CACHE : Hashtable<String, Typeface> = new Hashtable<>()
+get(manager : AssetManager, name : String) : Typeface
+get(assetManager : AssetManager) : Typeface

Gambar 142 CD TypefaceCache

### B.3 Class Diagram Untuk Super Admin Komunitas

#### B.3.1 Model

<b>Admin</b> (com::aldoapps::superadminkomunitasojekfinder::model)
-objectId : String
-name : String
-email : String
-isActive : String
-createdAt : Date
-updatedAt : Date
-activationStatus : String

Gambar 143 CD Admin



## B.3.2 User Interface

<b>AdminDetailActivity</b>
(com::aldoapps::superadminkomunitasojekfinder)
-mAdmin : Admin
#onCreate(savedInstanceState : Bundle) : void
-changeActivationStatus() : void
+onOptionsItemSelected(item : MenuItem) : boolean

**Gambar 144 CD AdminDetailActivity**

<b>AdminDetailFragment</b>
(com::aldoapps::superadminkomunitasojekfinder)
+ARG_ITEM_ID : String = "item_id"
-mAdmin : Admin
+AdminDetailFragment()
+onCreate(savedInstanceState : Bundle) : void
+onCreateView(inflater : LayoutInflater, container : ViewGroup, savedInstanceState : Bundle) : View

**Gambar 145 CD AdminDetailFragment**

<b>LoginActivity</b>
(com::aldoapps::superadminkomunitasojekfinder)
-mUserName : EditText
-mPassword : EditText
-mLogin : Button
-mProgressDialog : ProgressDialog
#onCreate(savedInstanceState : Bundle) : void
-doLogin() : void
-navigateToMainActivity(parseObject : ParseObject) : void
#onPause() : void

**Gambar 146 CD LoginActivity**



```

NewAdminActivity
(com::aldoapps::superadminkomunitasojekfinder)

-mEmail : EditText
-mName : EditText
-mPassword : EditText
-mPasswordAgain : EditText
-mAssignNewAdminBtn : Button
-mTextWatcher : TextWatcher
-mHintTextView : TextView

#onCreate(savedInstanceState : Bundle) : void
-createNewAdminJob() : void

```

Gambar 147 CD NewAdminActivity

```

SuperAdminApplication
(com::aldoapps::superadminkomunitasojekfinder)

+onCreate() : void

```

Gambar 148 CD SuperAdminApplication

### B.3.3 Utility Class

```

EncryptionUtil
(com::aldoapps::superadminkomunitasojekfinder)

+convertToMd5(s : String) : String

```

Gambar 149 CD EncryptionUtil

*(halaman ini sengaja dikosongkan)*



## LAMPIRAN C TEST CASE

Tabel 45 TC01 Sign Up

Test Case Code		TC-01				
Test Case Name		Sign Up				
No.	Aksi Test	Data Test	Hasil yang Diharapkan	Hasil Aktual	Sukses/Gagal	
1	Mendaftar dengan email yang belum ada	Username "aldo@yahoo.com" =	User terdaftar dalam aplikasi	Sistem mendaftarkan pengguna dan menampilkan halaman utama	Sukses	
		Email "aldo@yahoo.com" =				
		Password = "secret"				
2	Mendaftar dengan email yang sudah ada	Username "aldo@yahoo.com" =	User tidak dapat terdaftar dalam aplikasi	Sistem menampilkan <i>toast</i> bahwa username sudah digunakan	Sukses	
		Email "aldo@yahoo.com" =				
		Password = "secret"				



Tabel 46 TC02 Login

Test Case Code		TC-02			
Test Case Name		Login			
No.	Aksi Test	Data Test	Hasil yang Diharapkan	Hasil Aktual	Sukses/Gagal
1	Login dengan email yang belum ada	Email "aldo@yahoo.com" =	User masuk ke dalam aplikasi	Sistem memasukkan pemain ke dalam aplikasi	Sukses
		Password = "secret"			
2	Login dengan email yang belum terdaftar	Username "aldo@yahoo.com" =	User tidak masuk ke dalam aplikasi	Sistem menampilkan <i>toast</i> bahwa username dan password tidak sesuai	Sukses
		Password = "secret"			

**Tabel 47 TC03 Logout**

Test Case Code		TC-03			
Test Case Name		Logout			
No.	Aksi Test	Data Test	Hasil yang Diharapkan	Hasil Aktual	Sukses/Gagal
1	Logout dengan pengguna yang sudah login	email="aldo@yahoo.com"	Use keluar dari aplikasi	Sistem mengeluarkan pemain dari aplikasi dan menampilkan halaman <i>landing</i>	Sukses

**Tabel 48 TC04 Mengganti Nama dan No. HP**

Test Case Code		TC-04			
Test Case Name		Mengganti Nama dan No. HP			
No.	Aksi Test	Data Test	Hasil yang Diharapkan	Hasil Aktual	Sukses/Gagal



Test Case Code		TC-04			
Test Case Name		Mengganti Nama dan No. HP			
No.	Aksi Test	Data Test	Hasil yang Diharapkan	Hasil Aktual	Sukses/Gagal
1	Mengganti nama dan no. Telpn di halaman profil	name= "aldo ", phoneNumber="08996342553"	Nama dan no. hp pengguna berubah sesuai dengan data yang diupload	Sistem mengupload nama dan no.hp yang diinputkan user dan mengganti nama dan no.hp yang ditampilkan ke user	Sukses
2	Mengganti nama dan no. Telpn di halaman profil namun internet device pengguna tidak	name= "aldo ", phoneNumber="08996342553"	Nama dan no. hp tidak berubah	Sistem menampilkan toast “gagal mengganti nama atau no. hp”	Sukses



Test Case Code		TC-04			
Test Case Name		Mengganti Nama dan No. HP			
No.	Aksi Test	Data Test	Hasil yang Diharapkan	Hasil Aktual	Sukses/Gagal
	aktif				

**Tabel 49 TC05 Mengganti Profile Picture**

Test Case Code		TC-05			
Test Case Name		Mengganti Profile Picture			
No.	Aksi Test	Data Test	Hasil yang Diharapkan	Hasil Aktual	Sukses/Gagal
1	Mengganti foto profil dari kamera / galeri	email="aldo@yahoo.com", file="avatar.jpg"	Foto profil pengguna berubah sesuai dengan file yang <i>diupload</i>	Sistem mengupload file yang diinputkan user dan mengganti foto profil yang ditampilkan ke user	Sukses

Test Case Code		TC-05			
Test Case Name		Mengganti Profile Picture			
No.	Aksi Test	Data Test	Hasil yang Diharapkan	Hasil Aktual	Sukses/Gagal
2	Menekan tombol “cancel” ketika memilih foto	email="aldo@yahoo.com", file="avatar.jpg"	Foto profil user tidak berubah	Sistem menampilkan toast “gagal mengambil foto”	Sukses

Tabel 50 TC06 Bergabung ke Komunitas

Test Case Code		TC-06			
Test Case Name		Bergabung ke Komunitas			
No.	Aksi Test	Data Test	Hasil yang Diharapkan	Hasil Aktual	Sukses/Gagal
1	Mengisikan nama komunitas yang sesuai dengan komunitas yang ada	Nama komunitas="Sistem Informasi"	Pengguna terdaftar dalam komunitas	Aplikasi menampilkan <i>toast</i> berhasil bergabung dalam komunitas	Sukses



Test Case Code		TC-06			
Test Case Name		Bergabung ke Komunitas			
No.	Aksi Test	Data Test	Hasil yang Diharapkan	Hasil Aktual	Sukses/Gagal
2	Nama komunitas yang dicari tidak ditemukan	Nama komunitas="Sistem"	Pengguna tidak terdaftar dalam komunitas	Aplikasi menampilkan <i>toast</i> gagal bergabung dalam komunitas	Sukses

**Tabel 51 TC07 Keluar Komunitas**

Test Case Code		TC-07			
Test Case Name		Keluar Komunitas			
No.	Aksi Test	Data Test	Hasil yang Diharapkan	Hasil Aktual	Sukses/Gagal



Test Case Code		TC-07			
Test Case Name		Keluar Komunitas			
No.	Aksi Test	Data Test	Hasil yang Diharapkan	Hasil Aktual	Sukses/Gagal
1	Menekan tombol “remove community”	Nama komunitas=”Sistem Informasi”	Pengguna terhapus dalam komunitas	Aplikasi menampilkan <i>toast</i> berhasil keluar dalam komunitas	Sukses
2	Menekan tombol “remove community” namun pengguna belum bergabung dalam komunitas	Nama komunitas=””	Pengguna tidak terhapus dalam komunitas	Aplikasi menampilkan <i>toast</i> gagal keluar dalam komunitas	Sukses

**Tabel 52 TC08 Melihat Keberadaan Semua Pengguna**

Test Case Code		TC-08			
Test Case Name		Melihat Keberadaan Semua Pengguna			
No.	Aksi Test	Data Test	Hasil yang Diharapkan	Hasil Aktual	Sukses/Gagal
1	Menekan tombol pada “action bar”	Daftar pengguna yang terdaftar	Daftar pengguna berhasil terdaftar	Menampilkan lokasi semua pengguna	Sukses
2	Menekan tombol pada “action bar” namun internet atau GPS tidak aktif	Daftar pengguna yang terdaftar	Daftar pengguna tidak berhasil terdaftar	Peta yang ditampilkan tidak menunjukan lokasi siapapun	Sukses



Tabel 53 TC09 Memberikan Review dan Rating

Test Case Code		TC-09			
Test Case Name		Memberikan Review dan Rating			
No.	Aksi Test	Data Test	Hasil yang Diharapkan	Hasil Aktual	Sukses/Gagal
1	Mengisi review dan rating lalu menekan tombol ok	Confirm Object Id	Rating dan review objek terdaftar	Menampilkan <i>toast</i> bahwa pengirim rating dan review berhasil	Sukses
2	Mengisi salah satu review atau rating lalu menekan tombol ok	Confirm Object Id	Rating dan review objek terdaftar	Menampilkan <i>toast</i> bahwa pengirim rating dan review berhasil	Sukses



Test Case Code		TC-09			
Test Case Name		Memberikan Review dan Rating			
No.	Aksi Test	Data Test	Hasil yang Diharapkan	Hasil Aktual	Sukses/Gagal
3	Tidak mengisi salah satu review dan rating lalu menekan tombol ok	Confirm Object Id	Rating dan review objek tidak terdaftar	Menampilkan <i>toast</i> bahwa pengirim rating dan review gagal	Sukses

**Tabel 54 TC10 Membatalkan Transaksi**

Test Case Code		TC-10			
Test Case Name		Membatalkan Transaksi			
No.	Aksi Test	Data Test	Hasil yang Diharapkan	Hasil Aktual	Sukses/Gagal

Test Case Code		TC-10			
Test Case Name		Membatalkan Transaksi			
No.	Aksi Test	Data Test	Hasil yang Diharapkan	Hasil Aktual	Sukses/Gagal
1	Menekan item dalam daftar penumpang, lalu menekan tombol “yes”	Passenger Object Id	Passenger dan Rider Object terhapus	Menampilkan <i>toast</i> bahwa membatalkan transaksi berhasil	Sukses
2	Menekan item dalam daftar taksi motor, lalu menekan tombol “yes”	Rider Object Id	Rider Object terhapus	Menampilkan <i>toast</i> bahwa membatalkan transaksi berhasil	Sukses
3	Menekan item dalam daftar persetujuan, lalu menekan tombol “yes”	Confirm Object Id	Confirm Object terhapus	Menampilkan <i>toast</i> bahwa membatalkan transaksi berhasil	Sukses



**Tabel 55 TC11 Membuat Permintaan Tumpangan Baru**

Test Case Code		TC-11			
Test Case Name		Membuat Permintaan Tumpangan Baru			
No.	Aksi Test	Data Test	Hasil yang Diharapkan	Hasil Aktual	Sukses/Gagal
1	Mengisi lokasi keberangkatan dan tujuan lalu menekan tombol ok	Keberangkatan: “Keputih” Tujuan: “Stasiun Pasar Turi”	Passenger Object terdaftar	Menampilkan <i>toast</i> bahwa pengirim permintaan tumpangan berhasil	Sukses
2	Tidak lengkap mengisi lokasi keberangkatan dan lokasi tujuan lalu menekan tombol ok	Keberangkatan: “Keputih” Tujuan:” “	Passenger Object tidak terdaftar	Menampilkan <i>toast</i> bahwa pengirim permintaan tumpangan gagal	Sukses

Tabel 56 TC12 Menyetujui Penawaran Taksi Motor

Test Case Code		TC-12			
Test Case Name		Menyetujui Penawaran Taksi Motor			
No.	Aksi Test	Data Test	Hasil yang Diharapkan	Hasil Aktual	Sukses/Gagal
1	Menekan tombol “yes” pada salah satu daftar supir taksi motor	Rider Object Id	Passenger dan Rider Object dihapus, dan Confirm Object dibuat	Menampilkan <i>toast</i> bahwa konfirmasi persetujuan taksi motor berhasil	Sukses

Tabel 57 Membuat Penawaran Taksi Motor

Test Case Code		TC-13			
Test Case Name		Membuat Penawaran Taksi Motor			
No.	Aksi Test	Data Test	Hasil yang Diharapkan	Hasil Aktual	Sukses/Gagal



Test Case Code		TC-13			
Test Case Name		Membuat Penawaran Taksi Motor			
No.	Aksi Test	Data Test	Hasil yang Diharapkan	Hasil Aktual	Sukses/Gagal
1	Memilih harga yang disetujui, atau mengisi detail penawaran dan menekan tombol “yes”	Passenger Object Id	Rider Object terdaftar	Menampilkan <i>toast</i> bahwa pengirim penawaran taksi motor berhasil	Sukses
2	Memilih harga yang disetujui, atau mengisi detail penawaran dan menekan tombol “yes”. Namun daftar permintaan tumpangan sudah hilang	Passenger Object Id = “null”	Rider Object terdaftar	Menampilkan <i>toast</i> bahwa pengirim penawaran taksi motor gagal	Sukses

Tabel 58 Mengaktivasi dan Deaktivasi Pengguna

Test Case Code		TC-14			
Test Case Name		Mengaktivasi dan Deaktivasi Pengguna			
No.	Aksi Test	Data Test	Hasil yang Diharapkan	Hasil Aktual	Sukses/Gagal
1	Memilih pengguna yang ingin diaktivasi/deaktivasi, lalu menekan <i>floating action button</i>	UserCommunity	UserCommunity isActive="yes" / "no"	Menampilkan <i>toast</i> bahwa perubahan status aktivasi/deaktivasi pengguna berhasil	Sukses

Tabel 59 TC15 Membuat Admin Baru

Test Case Code		TC-15			
Test Case Name		Membuat Admin Baru			
No.	Aksi Test	Data Test	Hasil yang Diharapkan	Hasil Aktual	Sukses/Gagal



Test Case Code		TC-15			
Test Case Name		Membuat Admin Baru			
No.	Aksi Test	Data Test	Hasil yang Diharapkan	Hasil Aktual	Sukses/Gagal
1	Mengisikan semua form email, name dan password lalu menekan tombol “ok”	Email= <a href="mailto:aldo@yahoo.com">aldo@yahoo.com</a> , name=”aldo” password=”123”	CommunityAdmin terdaftar	Menampilkan <i>toast</i> bahwa pendaftaran admin baru berhasil	Sukses
2	Tidak lengkap dalam mengisikan form email, name dan password lalu menekan tombol “ok”	Email=’’, name=’’, password=”123”	CommunityAdmin tidak terdaftar	Menampilkan <i>toast</i> bahwa pendaftaran admin baru gagal	Sukses

Tabel 60 TC16 Mengaktivasi dan Deaktivasi Admin Komunitas

Test Case Code		TC-16			
Test Case Name		Mengaktivasi dan Deaktivasi Admin Komunitas			
No.	Aksi Test	Data Test	Hasil yang Diharapkan	Hasil Aktual	Sukses/Gagal
1	Memilih item admin yang ingin diaktivasi/deaktivasi, lalu menekan <i>floating action button</i>	CommunityAdmin Object Id	CommunityAdmin isActive="yes"/"no"	Menampilkan <i>toast</i> bahwa perubahan status aktivasi/deaktivasi admin berhasil	Sukses